Nilson Barros Santos

Desempenho das Simulações AR e MA na plataforma Livre R

Brasil

2015

Nilson Barros Santos

Desempenho das Simulações AR e MA na plataforma Livre R

Monografia apresentada ao Departamento de Ciências Atuariais e Estatística - DECAT da Universidade Federal de Sergipe - UFS, como resultado do Trabalho de Conclusão do Curso de Estatística Bacharelado e um dos pré-requisitos para obtenção do Título de Bacharel. Orientador: Daniel Francisco Neyra Castañeda.

Universidade Federal de Sergipe - UFS

Departamento de Ciências Atuariais e Estatística - DECAT

Bacharelado em Estatística

Brasil

2015

Brasil, 2015-

Este trabalho é dedicado a todos aqueles que de alguma forma contribuíram nessa longa caminhada.

Agradecimentos

Agradeço aos dedicados Professores do Departamento de Estatística da UFS, que sempre estiveram à disposição para sanar todas as nossas dúvidas. Em especial ao Msc. Daniel Castañeda, pela dedicação, orientação e lições de vida, ensinadas durante esses quatro anos. Aos meus familiares pelo incentivo e inspiração durante toda essa jornada. E, por fim, a todos os colegas de curso, em especial aqueles que mesmo durante esse longo e tortuoso caminho permaneceram unidos pelo objetivo em comum.

À todos vocês, meu muito obrigado!

"...tenha coragem de seguir o seu próprio coração e a sua intuição. Eles de alguma maneira já sabem o que você realmente quer se tornar. Todo o resto é secundário".

(Steve Jobs, 2005)

Resumo

O presente trabalho tem como objetivo avaliar o desempenho das simulações de modelos Autorregressivos (AR) e de Médias Móveis (MA) na Plataforma Livre R. Para tanto, utilizamos um conjunto de procedimentos (funções e comandos) do Software Livre R, gerando séries simuladas para diferentes cenários. Foram simuladas, através de algoritmo, séries temporais para os modelos: AR(1), AR(2), MA(1) e MA(2), variando os valores dos parâmetros de cada modelo proposto, o tamanho das séries (foram simuladas séries com $n = 10$,$n = 20$, $n = 100$, $n = 500$ e $n = 1000$ observações), bem como a quantidade de séries geradas para cada valor dos parâmetros, sendo simuladas entre 90 a 9000 séries em cada execução do algoritmo. As séries temporais simuladas através da função *arima.sim* foram analisadas de forma automática pela função *auto.arima* e, a partir da observação da frequência das séries temporais geradas, em consonância com o modelo inicialmente proposto, pode-se concluir que as séries simuladas nem sempre correspondem ao modelo inicialmente proposto. Especialmente para simulações de séries com n pequeno, ou seja, com poucas observações. Para melhor compreensão do trabalho abordamos ainda os principais conceitos de séries temporais, da modelagem proposta por Box e Jenkins para os processos estocásticos - AR(p) e MA(q) e, finalmente os principais conceitos relacionados à simulação estatística (Método de Simulação de Monte Carlo).

Palavras-chaves: Algoritmo. Monte Carlo. Séries Temporais.

Abstract

This study aims to evaluate the performance of Autoregressive (AR) model simulations and Moving Average (MA) in the Free Platform R. Therefore, we use a set of procedures (functions and commands) in Free Software R, generating simulated series for different scenarios. Were simulated by algorithm, time series models: AR (1), AR (2), MA (1) and MA (2) varying the parameter values of each proposed model, the size of the series (were simulated series with $n = 10$, $n = 20$, $n = 100$, $n = 500$ and $n = 1,000$ observations) as well, the amount of generated series for each value of the parameters is simulated between 90-9000 series in each run of the algorithm. The time series simulated by the *arima.sim* were analyzed automatically using by the function *auto.arima* and through observation the frequency of the generated time series, in line with the initially proposed model, conclude that the simulated series do not always correspond to the model originally proposed. Especially for a series of simulations with n small, i.e. with few observations. To better understand the work still approach the key concepts of time series modeling proposed by Box and Jenkins for stochastic processes - AR (p) and MA (q) and, finally the main concepts related to statistical simulation (Simulation Method Monte Carlo).

Key-words: Algorithm. Monte Carlo. Time Series.

Sumário

Lista de ilustrações

Lista de tabelas

Introdução

O uso de computadores em Matemática e Estatística, possibilitou o surgimento de técnicas para estudar problemas que de outro modo seriam insolúveis. “Computação estatística” é o ramo da matemática que diz respeito às técnicas que envolvem diretamente a aleatoriedade, ou em que a aleatoriedade é usada como parte de um modelo matemático [...] (VOSS, 2011).

Segundo Gomes (1994) muitas vezes na busca da solução de problemas, nas diversas áreas do conhecimento humano (física, química, matemática etc.) surge a impossibilidade de obtermos tal solução analítica ou numericamente. Tal fato pode ocorrer por diversos e variados motivos, podendo ser aqui citado, por exemplo, o custo operacional de se obter dados reais para a realização de determinado estudo. Para Voss (2011) uma das ideias mais importantes da computação estatística, é que em muitos casos as propriedades de um determinado modelo estocástico só podem ser observados experimentalmente, fazendo-se uso então do computador para gerar muitos exemplos ao acaso do modelo, para então analisar a amostra resultante. Usualmente e, principalmente no meio acadêmico-científico são utilizadas técnicas de simulação que num sentido amplo recebe a denominação de *Simulação de Monte Carlo* ou *Algoritmo de Monte Carlo*.

Diversos são os softwares atualmente utilizados para gerar simulações, sendo o Software Livre, R-Project, um deles. Rotinas específicas desse software podem ser utilizadas para simular dados de determinada distribuição de probabilidade, bem como, armazenar sequências de números pseudo aleatórios que podem ser representadas por meio de séries temporais.

Números pseudo aleatórios são números que podem ser deterministicamente gerados em computador, mas sobre o ponto de vista estatístico, podem ser considerados aleatórios como, por exemplo, as observações independentes de uma Distribuição Uniforme (0,1)(GOMES, 1994).

A questão que podemos levantar a respeito do tema em epígrafe é se os modelos gerados são sempre correspondentes aos modelos teóricos pretendidos e qual a eficiência da simulação de tais modelos. Nesse trabalho iremos fazer uso de um conjunto de algoritmos para gerar séries temporais simuladas através de comandos específicos do Software R (comando *arima.sim*, por exemplo), especificando todos os parâmetros do modelo teórico a ser gerado, variando o tamanho (n) da amostra gerada, os valores dos parâmetros dos modelos estudados e a quantidade de repetições.

Os modelos Auto-regressivos são de suma importância no estudo de séries temporais, sendo bastante comuns em estudos econométricos. Segundo Ehlers (2003) “processos AR

podem ser usados como modelo se for razoável assumir que o valor atual de uma série temporal depende de seu passado imediato mais um erro aleatório". Já o processo de Médias Móveis, denotado como MA(q), segundo Ehlers (2003), é assim definido quando Y_t é gerada por uma média ponderada dos erros aleatórios de q períodos passados. Sendo que o entendimento desses dois modelos, tem uma importância fundamental no estudo da modelagem Box-Jenkins de séries temporais mais complexas, tanto estacionárias (Modelos Autorregressivos Médias Móveis) quanto as não estacionárias (Modelos Autorregressivos integrados-Médias Móveis) e, conforme já dito, o estudo de alguns fenômenos só é possível através de simulação computacional.

Por conseguinte, pretendemos testar o desempenho das simulações dos Modelos de séries temporais do tipo AR(p) (Auto-regressivos) e MA(q) (por Média Móveis) da plataforma livre R, verificando o quanto os comandos e rotinas deste Software são precisos na geração dos modelos em epígrafe.

Para o desenvolvimento deste trabalho um dos principais problemas encontrados está relacionado ao custo computacional, uma vez que a geração e a análise de séries temporais com um número elevado de observações, em computadores pessoais é uma tarefa ainda árdua, demandando bastante tempo de processamento.

O entendimento deste artigo requer do leitor apenas o conhecimento básico de estatística e probabilidade e, por ser um texto que trata de métodos computacionais, aqueles que possuem habilidades em programação poderão tirar maior proveito do artigo em tela. Devemos ressaltar, no entanto, que o entendimento da parte principal do estudo não requer o conhecimento de qualquer linguagem de programação e, esperamos ao final deste estudo ter contribuído de alguma forma para a implementação de melhorias no Software estatístico R-Project.

Para melhor compreensão do trabalho iremos abordar todos os temas citados acima, explicando os principais conceitos de séries temporais, da modelagem proposta por Box e Jenkins para os processos estocásticos - AR(p) e MA(q) e, finalmente os principais conceitos relacionados à simulação estatística (Método de Simulação de Monte Carlo).

Revisão de Literatura

1.1 Séries Temporais

1.1.1 Definição

Como preâmbulo iremos iniciar nosso trabalho trazendo alguns conceitos e definições relativos à Análise de Séries Temporais.

Para Morettin e Toloi (1981) uma Série Temporal é qualquer conjunto de observações ordenadas no tempo, e como exemplo de séries temporais temos:

1. estimativas trimestrais do PIB;
2. valores diários da temperatura de Aracaju;
3. valores diários da Bolsa de Valores de São Paulo;
4. quantidade mensal de chuvas na cidade de Salvador;
5. valores mensais das vendas de automóveis no Brasil;
6. um registro de marés no Porto de Santos (MORETTIN; TOLOI, 1981).

> A partir dos exemplos acima pode-se classificar as séries temporais em discretas ou contínuas. As séries obtidas de 1 a 5 são discretas, enquanto que em 6 temos um exemplo de série contínua. Importante notar que, por diversas vezes uma série temporal discreta é obtida através da amostragem de uma série temporal contínua em intervalos de tempos iguais, Δ. Portanto, a série do item 6 para ser analisada é necessário que esta seja amostrada (em intervalos de tempo de uma hora, por exemplo), convertendo a série contínua, observada no intervalo $[0, T]$, em uma série discreta com N pontos, onde $N = [\frac{T}{\Delta}]$ (MORETTIN; TOLOI, 1981).

Para Morettin e Toloi (1981) os objetivos da análise de séries temporais são:

1. investigar o mecanismo gerador da série temporal;
2. fazer previsões de valores futuros da série;
3. descrever apenas o comportamento da série.

Para Morettin e Toloi (1981) há dois enfoques básicos na análise de séries temporais. Um diz respeito ao domínio temporal e os modelos propostos são os *paramétricos* e o

outro diz respeito a análise no domínio das frequências e os modelos propostos são *não paramétricos*.

Entendemos porém que a motivação principal da análise de séries temporais é a modelagem, com o intuito de fazer previsões. E, para tanto, se faz necessário entender o processo gerador da série, ou seja seu processo estocástico.

1.1.2 Processo Estocástico

"Matematicamente um processo estocástico pode ser definido como uma coleção de variáveis aleatórias ordenadas no tempo e definidas em um conjunto de pontos T, que pode ser contínuo ou discreto" (EHLERS, 2003). Ou seja, processos estocásticos são modelos probabilísticos que podem ser atribuídos, adequadamente, para dados de séries temporais.

Para um melhor entendimento do conceito de processo estocástico e série temporal, observe o gráfico abaixo:

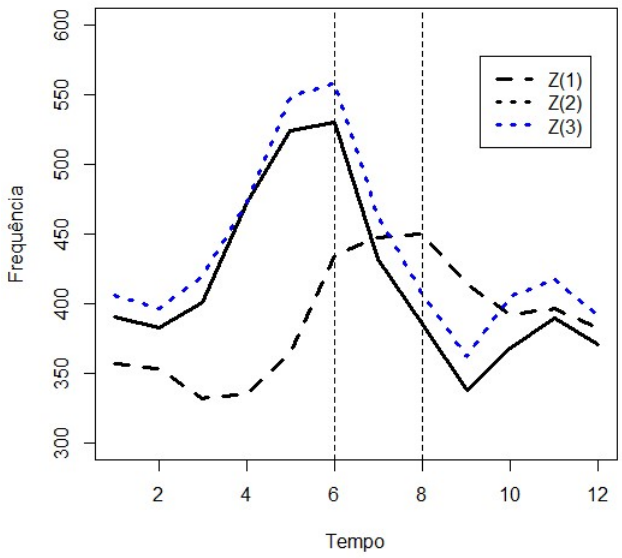

Figura 1 – Exemplo de Processo Estocástico

Fonte: Elaborado pelo Autor

Segundo Morettin e Toloi (1981) cada curva são *trajetórias* do processo físico que está sendo e observado. Podemos então definir processo estocástico como sendo o conjunto de todas as possíveis trajetórias que poderíamos observar e, portanto cada trajetória é chamada de *série temporal*.

Portanto, denotando a variável aleatória $Z(t)$ para o caso contínuo (para $-\infty < t < \infty$), e por Z_t para o caso discreto (com $t = 0, \pm1, \pm2, ...$), temos no gráfico acima três

trajetórias, em que cada observação da variável aleatória z_t no tempo t pode ser definida como uma realização do processo estocástico.

Então, designando $Z(6)_1$ como o valor de determinada observação no instante $t = 6$ como sendo a primeira observação, teremos um número real, para o segundo instante, teremos outro número real, $Z(8)_2$ e assim sucessivamente. Para cada t fixo, teremos os valores de uma variável aleatória Z_t, que terá certa distribuição de probabilidade (MORETTIN; TOLOI, 1981).

> Na realidade, o que chamamos de série temporal, é uma parte de uma trajetória, dentre muitas que poderiam ter sido observadas. Em algumas situações (como em Oceanografia, por exemplo), quando temos dados experimentais, é possível observar algumas trajetórias do processo em consideração, mas na maioria dos casos (como em Economia ou Astronomia), quando não é possível fazer experimentações, temos uma só trajetória para análise (MORETTIN; TOLOI, 1981)

Uma maneira de descrever um processo estocástico é através da distribuição de probabilidade conjunta de $Z(t_1), ..., Z(t_k)$ para qualquer conjunto de tempos $t_1, ..., t_k$ e qualquer valor de k. Sendo, contudo uma tarefa bastante complicada (MORETTIN; TOLOI, 1981).

Conforme descrito por Ehlers (2003) é bastante comum descrever um processo estocástico através das funções média, variância e autocovariância. Tais funções são definidas para o caso contínuo da forma descrita abaixo, podendo ser aplicadas, por similaridade, ao caso discreto:

$$\mu_t = E[Z(t)];$$

$$\sigma^2(t) = Var[Z(t)];$$

$$\gamma(t_1, t_2) = E[Z(t_1) - \mu(t_1)][Z(t_2) - \mu(t_2)].$$

Pode-se notar, pelas expressões acima que a função de variância é um caso particular da função de autocovariância, quando $t_1 = t_2$.

1.1.3 Outras noções importantes

Passaremos agora a estudar alguns conceitos importantes sobre a análise de séries temporais. Importante destacar que por não ser o objetivo principal de nosso trabalho, iremos apenas aqui expor algumas noções básicas, sem no entanto nos aprofundarmos.

1.1.3.1 Decomposição

Segundo o modelo clássico, as séries temporais podem ser compostas por quatro padrões a seguir:

a) Tendência (T): segundo Ehlers (2003) é o comportamento observado ao longo da série (decrescimento, crescimento), devendo ser considerado a longo prazo, podendo ter diversas causas, como por exemplo o crescimento demográfico, controle de natalidade, etc, ou seja, qualquer aspecto que interfira na variável em estudo a longo prazo.

b) Variações cíclicas ou ciclos (C): para Ehlers (2003) são variações ocorridas na série com periodicidade superior a um ano, podendo ter como causa, variações econômicas, fenômenos climáticos (como El Niño ou La Niña).

c) Variações Sazonais ou Sazonalidade (S): diferencia-se das variações cíclicas por interferir na série em períodos menores que um ano. Um exemplo clássico temos as estações do ano, que influenciam em experimentos afetados por mudanças no clima.

d) Variação Irregular ou Erro Aleatório (ϵ_i): são flutuações das quais não se é possível estabelecer um padrão, ou seja, é a parte aleatória do modelo, também chamado de ruído branco, podendo ser causado por catástrofes naturais, por exemplo (EHLERS, 2003).

Portanto, a partir das características acima citadas, podemos definir um modelo aditivo de série temporal através da seguinte expressão:

$$Z_t = T + C + S + \epsilon_i \tag{1.0}$$

Onde:

$T = Tendência$;

$C = Ciclo$;

$S = Sazonalidade$;

$\epsilon_i = Erro\ aleatório$;

Devemos salientar no entanto, que nem sempre a série temporal vai apresentar todos os componentes acima descritos.

Podemos verificar como cada componente acima afeta as séries temporais nos gráficos a seguir:

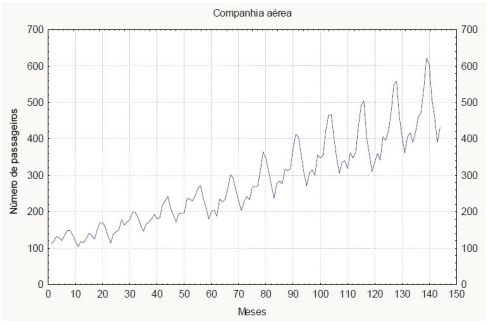

Figura 2 – Série com Tendência e Sazonalidade

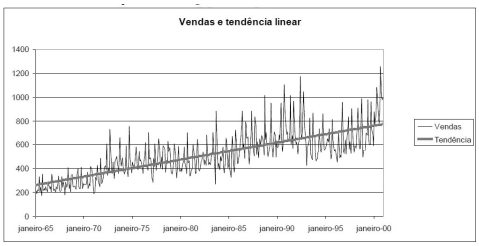

Figura 3 – Série com Têndencia de Crescimento

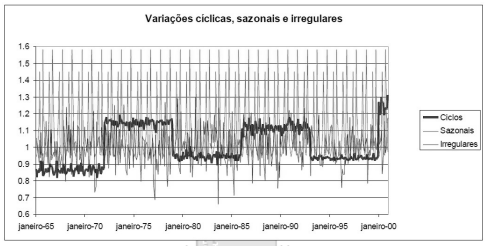

Figura 4 – Série com todos os componentes

Fonte:(REIS, 2014)

1.1.3.2 Estacionariedade

Uma das suposições mais frequentes que se faz a respeito de uma série é a de que ela é estacionária, ou seja, se desenvolve no tempo aleatoriamente ao redor de uma média constante, refletindo alguma forma de equilíbrio estável (MORETTIN; TOLOI, 2006 apud OLIVEIRA, 2012).

Na maioria dos procedimentos de análise estatísticas de séries temporais, supõe-se que estas sejam estacionárias. Com isso, caso a série não seja estacionária, será necessário transformar os dados originais, até obter uma série estacionária (MORETTIN; TOLOI, 2006, p. 5).

Para Stock e Watson (2004 apud OLIVEIRA, 2012), uma série de dados é dita estacionária, quando suas variáveis não apresentam tendências, sendo estáveis ao longo do tempo. Para as séries temporais é importante que as variáveis sejam estacionárias ou passíveis de se tornarem estacionárias. Sendo essa uma característica fundamental para previsão com base na regressão de séries temporais.

Atualmente temos diversos testes para verificar a estacionariedade das séries temporais, dos quais podemos citar, o Teste da Raiz Unitária, Dickley-Fuller e Dickley Fuller Aumentado, Teste de Phillips - Perron, entre outros.

1.1.3.3 Invertibilidade

Quanto a essa característica das séries temporais iremos analisar apenas as condições de invertibilidade dos dois modelos estudados nesse trabalho, ou seja, os modelos Autorregressivo AR(p) e de Médias Móveis MA(q). Segundo Bezerra (2006) toda Série Temporal Autorregressiva AR(p) é invertível, não existindo qualquer condição para garantir sua invertibilidade.

Segundo Bezerra (2006) o modelo MA(1) é dito invertível quando pode ser "invertido" (transformado) em AR(∞). Para um modelo geral MA(q), definimos o polinômio característico MA como:

$$\theta(x) = 1 - \theta_1 x - \theta_2 x^2 - ... - \theta_q x^q \tag{1.1}$$

e a sua equação característica é:

$$1 - \theta_1 x - \theta_2 x^2 - ... - \theta_q x^q = 0 \tag{1.2}$$

Pode-se então, demonstrar que o modelo MA(q) é invertível, e existirão constantes π_j, tal que:

$$Z_t = \sum_{j=1}^{\infty} \pi_j Z_{t-j} + a_t \tag{1.3}$$

se e somente se as raízes da equação característica MA execede a unidade em valor absoluto. (BEZERRA, 2006)

1.2 Modelos de Box-Jenkins para série estacionária

1.2.1 Definição

> Para Gujarati (2006) a questão fundamental quando estudamos séries temporais obviamente é: olhando para uma série temporal, como a do PIB dos Estados Unidos, descobrir se ela segue um processo autorregressivo puro (e sendo assim qual o valor de p) ou um processo MA puro (e, nesse caso, qual é o valor de q) ou se se trata de um processo ARMA (e quais são os valores de p e q) [...]. O Método Box-Jenkins é útil para responder a essa indagação.

Segundo Gujarati (2006) O método de Box-Jenkins, consiste em quatro etapas. Contudo antes de entrarmos no detalhamento do método, em epígrafe, faz-se necessário dizer que, devido ao objetivo desse trabalho nos limitaremos ao estudo dos modelos propostos por Box-Jenkins apenas para série estacionária ARMA (autorregressivos e de médias móveis), com maior atenção aos dois modelos que o compõe, MA(q)(Modelo Médias Móveis) e AR(p) (Autorregressivos).

Passamos então ao detalhamento do Método de Box-Jenkins:

1. **Identificação.** Ou seja, encontrar os valores adequados de p e q. Para isso podemos contar com o auxílio do *correlograma* e o *correlograma parcial*;

2. **Estimação.** Após identificarmos os valores adequados de p e q, passamos então a estimação dos parâmetros dos termos autorregressivos e de médias móveis incluídos no modelo, podendo ser utilizado o método dos mínimos quadrados ordinários, ou métodos de estimação não lineares;

3. **Verificação de Diagnóstico.** Depois de escolhido um determinado modelo ARMA e de estimados seus parâmetros, devemos verificar se o modelo escolhido se ajusta razoavelmente aos dados, sendo necessário, muitas vezes, que o analista tenha habilidade para escolher o modelo adequado;

4. **Previsão.** Em muitos casos as previsões obtidas são mais confiáveis que aquelas obtidas por outros métodos econométricos, sendo essa uma das principais razões do sucesso dos Modelos Box-Jenkins.

1.2.1.1 Autocorrelação

Segundo Ehlers (2003) a Autocorrelação é uma importante ferramenta utilizada para identificar as propriedades de uma série temporal, consistindo em uma série de quantidades chamadas *coeficientes de autocorrelação amostral*.

A ideia é semelhante ao coeficiente de correlação usual, isto é, para n pares de observações das variáveis x e y o coeficiente de correlação amostral é dado por:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{1.4}$$

Contudo, como estamos falando de séries temporais, temos que modificar a fórmula acima para medir a correlação entre as observações de uma mesma variável em diferentes horizontes de tempo, ou seja, correlações entre observações defasadas $1, 2, ...n$ períodos de tempo (EHLERS, 2003).

Então, considerando as n observações $x_1, ..., x_n$ de uma série temporal discreta podemos formar os pares $(x_1, x_2), ..., (x_{n-1}, x_n)$. Considerando $x_1, ..., x_{n-1}$ e $x_2, ..., x_n$ como duas variáveis o coeficiente de correlação entre x_t e x_{t+1} é dado por:

$$r = \frac{\sum_{t=1}^{n-1}(x_t - \overline{x}_1)(x_{t+1} - \overline{x}_2)}{\sqrt{\sum_{t=1}^{n-1}(x_t - \overline{x}_1)^2 \sum_{t=1}^{n-1}(x_{t+1} - \overline{x}_2)^2}} \tag{1.5}$$

Onde as médias amostrais são:

$$\overline{x}_1 = \frac{\sum_{t=1}^{n_1} x_t}{(n_1)} \text{ e } \overline{x}_2 = \frac{\sum_{t=2}^{n_2} x_t}{(n_2)}$$

Como o coeficiente r_1 mede as correlações entre observações sucessivas ele é chamado de coeficiente de autocorrelação ou coeficiente de correlação serial (EHLERS, 2003). E, assim como, o coeficiente de correlação, usualmente calculado, as autocorrelações são adimensionais e $-1 < r_k < 1$. Conforme (EHLERS, 2003) normalmente é mais comum calcular, a priori, os coeficientes de autocovariância c_k, definidos por analogia com a fórmula usual de covariância, isto é:

$$c_k = \sum_{t=1}^{n-k}(x_t - \overline{x})(x_{t+k} - \overline{x})/n$$

Logo os coeficientes de autocorrelação são então obtidos como $r_k = c_k/c_0$.

1.2.1.2 O Correlograma

> Um gráfico com os k primeiros coeficientes de autocorrelação como função de k é chamado de correlograma e pode ser uma ferramenta poderosa para identificar características da série temporal. Porém, isto requer uma interpretação adequada do correlograma, isto é, devemos associar certos padrões do correlograma com determinadas características de uma série temporal. Esta nem sempre é uma tarefa simples [...] (EHLERS, 2003).

Uma questão que pode ser respondida a partir do estudo do correlograma é se uma série temporal é aleatória ou não. Para uma série completamente aleatória os valores defasados são não correlacionados e portanto espera-se que $r_k \approx 0, k = 1, 2, ...$ (EHLERS, 2003).

A Figura(5) exibe o gráfico de uma série temporal com 100 observações i.i.d. geradas por meio computacionais e a Figura(6) seu correlograma. Neste caso os limites de confiança de 95% são aproximadamente $\pm 2/\sqrt{100} = \pm 0,2$. (EHLERS, 2003)

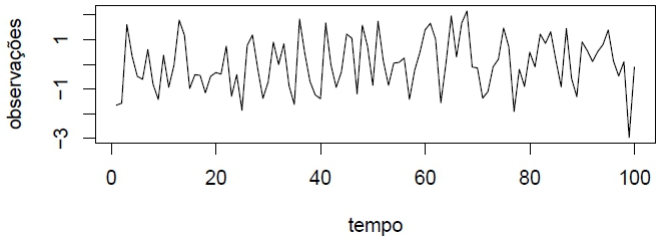

Figura 5 – Série Temporal Aleatória

Fonte: Ehlers (2003)

Podemos notar, no gráfico do correlograma (figura 6) que 2 dentre as 20 primeiras autocorrelações estão ligeiramente fora destes limites.

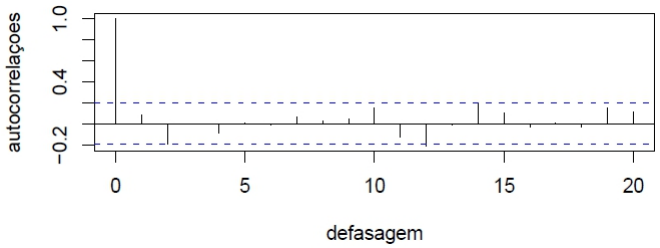

Figura 6 – Correlograma

Fonte: Ehlers (2003)

No entanto isto ocorre em defasagens aparentemente arbitrárias (12 e 14), e podemos concluir que não há evidências para rejeitar a hipótese de que as observações são independentes (EHLERS, 2003).

Esse é um exemplo de como podemos utilizar o correlograma para identificar características de uma série temporal. Abaixo vemos outros padrões de correlogramas, podendo ser analisado, por exemplo, qual o modelo mais adequado e quanto a estacionaridade da série.

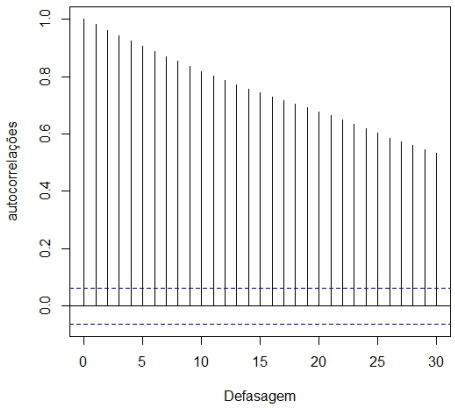

Figura 7 – (a) - Série com decaimento lento (não estacionária)

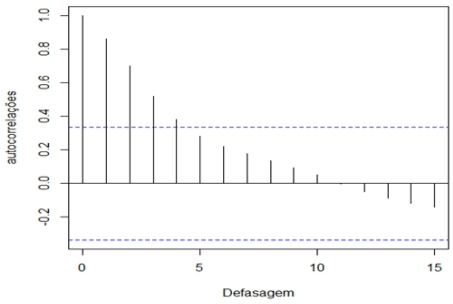

Figura 8 – (b) - Com decaimento lento e mudança estrutural

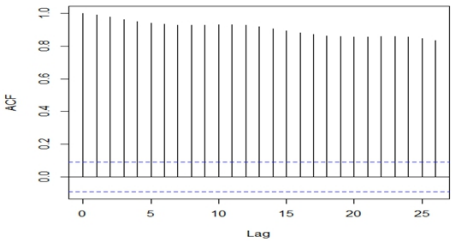

Figura 9 – (c)- Com decaimento extremamente lento

Fonte: Elaborado pelo Autor

1.3 Modelos Lineares Estacionários (AR e MA)

De acordo com a proposta desse trabalho, passaremos a analisar dois dos 3 (três) modelos lineares estacionários existentes: o modelo Autorregressivo (AR(p)) e o de Média Móveis (MA(q)).

Devemos ressaltar que o terceiro modelo, não analisado neste estudo, é uma combinação dos outros citados acima (ARMA(p, q)) e é uma proposição de Box-Jenkins para séries que encerram, tanto caracteríscas de AR quanto de MA.

1.3.1 Modelos Autorregressivos

Considere o modelo dado pela expressão abaixo:

$$(Z_t - \delta) = \alpha_1(Z_{t-1} - \delta) + \epsilon_t \tag{1.6}$$

Onde δ é a média de Y e ϵ_t é um termo de erro aleatório não correlacionado, com média zero e variância constante σ^2 (isto é, trata-se de um ruído branco), então dizemos que Z_t segue um processo *autorregressivo estocástico de primeira ordem* ou $AR(1)$ (GUJARATI, 2006).

> Aqui o valor de Z no período t depende de seu valor no período anterior e de um erro aleatório; os valores de Z são expressos como desvios de seu valor médio. Ou seja, para este modelo o valor previsto de Z no período t é uma proporção ($= \alpha_1$) de seu valor no período $(t-1)$, mais um choque ou distúrbio aleatório no período t; os valores de Z são expressos em torno de seu valor médio (GUJARATI, 2006).

Se considerarmos o modelo da expressão abaixo:

$$(Z_t - \delta) = \alpha_1(Z_{t-1} - \delta) + \alpha_2(Z_{t-2} - \delta) + \epsilon_t \tag{1.7}$$

por conseguinte, temos então que Z_T segue um processo *autorregressivo de segunda ordem*, ou $AR(2)$, ou seja o valor previsto de Z depende de seu valor nos dois períodos anteriores.

Para o caso geral a expressão fica:

$$(Z_t - \delta) = \alpha_1(Z_{t-1} - \delta) + \alpha_2(Z_{t-2} - \delta) + ... + \alpha_p(Z_{t-p} - \delta) + \epsilon_t \tag{1.8}$$

ou seja, a expressão acima representa um processo *autorregressivo de ordem p* ou $AR(p)$.

1.3.2 Modelos de Médias Móveis

Considere o seguinte modelo abaixo:

$$Z_t = \mu + \beta_0\epsilon_t + \beta_1\epsilon_{t-1} \tag{1.9}$$

Onde, μ é uma constante, e ϵ, é o termo de erro estocástico, ou ruído branco. Nesse modelo Z no período t é igual a uma constante mais uma média móvel dos termos de erro presentes e passados. Assim, neste caso, dizemos que Y segue um processo de *média móvel de primeira ordem*, ou $MA(1)$ (GUJARATI, 2006).

Mas seguindo a expressão acima, temos:

$$Z_t = \mu + \beta_0\epsilon_t + \beta_1\epsilon_{t-1} + \beta_2\epsilon_{t-2} \tag{1.10}$$

ou seja, um processo $MA(2)$. E de forma geral temos:

$$Z_t = \mu + \beta_0\epsilon_t + \beta_1\epsilon_{t-1} + \beta_2\epsilon_{t-2} + ... + \beta_q\epsilon_{t-q} \tag{1.11}$$

por conseguinte, um processo $MA(q)$. Portanto, podemos concluir que um processo de média móvel é simplesmente uma combinação linear de termos de erro de um ruído branco.

1.4 Simulação Estatística

Seguindo então a proposta deste trabalho, iremos abordar, de forma mais detalhada, todos os aspectos que entendemos ser relevantes, relacionados à Simulação Estatística.

1.4.1 Introdução a Simulação

O uso de computadores em matemática e estatística, abriu uma ampla gama de técnicas para estudar problemas que de outro modo seriam insolúveis, assim como analisar grandes conjuntos de dados. (VOSS, 2011).

Para Abreu e Rangel (1999) simular uma realidade é uma forma de ampliar o conhecimento e avançar no aprimoramento dos produtos. Nesse contexto, temos como exemplo o uso da simulação na indústria que, através da simulação de réplicas, pode testar as características de produtos, porém com um menor custo.

> A técnica de Simulação computacional consiste em estabelecer um modelo capaz de descrever ou representar o problema real a ser submetido à manipulação "experimental" em um computador. Em outras palavras, Simulação Computacional consiste em conduzir "experimentos" em um computador, envolvendo relações de conteúdo lógico e matemático, necessários à descrição do comportamento da estrutura de um problema real, em períodos de tempos bem definidos (ABREU; RANGEL, 1999)

Para Abreu e Rangel (1999) o uso da Simulação tem os seguintes objetivos principais:

i. compreender o funcionamento de um processo;

ii. prever o desempenho de determinado processo;

iii. prever o comportamento;

iv. dominar e controlar a evolução do sistema real, através de estratégias a serem adotadas dentro das restrições impostas pela realidade.

Para Gomes (1994) muitas vezes a procura para solucionar problemas (em Matemática, Estatística, Economia, Saúde, etc.) pode ser impossibilitada por diversos motivos, e em muitos casos, não há como se obter a solução de forma analítica ou mesmo numericamente. É então que se utiliza as técnicas de Simulação, designado em sentido *lato* Método de Monte Carlo.

Para Gomes (1994), podemos definir Simulação como sendo um "jogo" efetuado em computador, que envolve o uso de Números *Pseudo Aleatórios*, gerados de forma determinística, por softwares específicos, que sob o ponto de vista estatístico podem ser considerados como sendo *Números Aleatórios*.

Podemos esquematizar uma simulação estatística da seguinte forma:

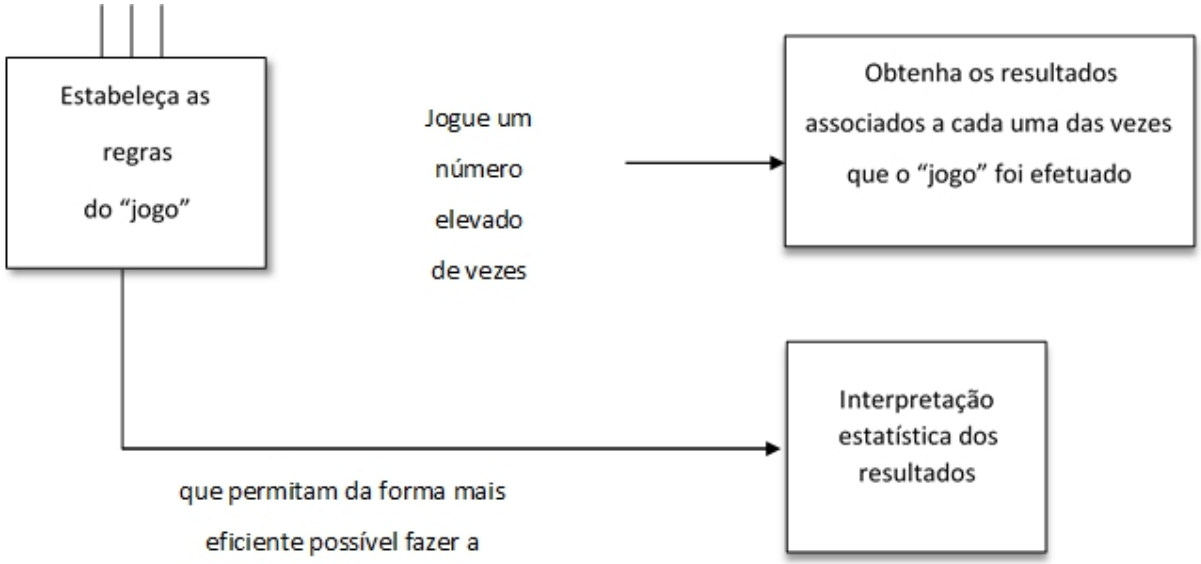

Figura 10 – Esquema de uma simulação

Fonte:(GOMES, 1994)

1.4.2 Números Aleatórios

Os números aleatórios tem uma importância fundamental na Simulação Estatística. Para Voss (2011) existem duas classes fundamentalmente diferentes de métodos para gerar números aleatórios:

a) os verdadeiros números aleatórios; gerados usando algum fenômeno físico que é aleatório. Caro e dispendioso, envolve o uso de hardwares avançados (para experimentos

na física quântica por exemplo), ou muito tempo (como o lançamento de um dado repetidas vezes);

b) números pseudo-aleatórios; são gerados por programas de computador. Embora estes métodos sejam rápidos e eficazes, um dos problemas é que os programas são inerentemente deterministas, e, portanto, não produzem números verdadeiramente aleatórios.

O mais usual é termos experimentos com números *pseudo-aleatórios*, com sua geração variando de acordo com o software utilizado.

Para Voss (2011) um gerador de números pseudo-aleatórios (GNPA) é um algorítmo que gera uma sequência de números que pode ser usada em substituição de uma sequência de números aleatórios independentes e identicamente distribuídos.

1.4.2.1 Exemplo de GNPA

Um exemplo (simples) de um gerador de números pseudo-aleatório é dado no seguinte algoritmo:

Algorítmo LCG (Gerador Congruencial Linear)
entrada:
$m > 1$ (*o módulo*),
$a \in \{1, 2, ..., m-1\}$ (*o multiplicador*),
$c \in \{0, 1, ..., m-1\}$ (*o incremento*);
$X_0 \in \{0, 1, ..., m-1\}$ (*a semente*);
saída:
a sequência $X_1, X_2, X_3, ...$ de números pseudo-aleatórios.

for $n = 1, 2, 3, ...,$ **do**
$X_n \leftarrow (aX_{n-1} + c)\ mod\ m$
output X_n
end for (VOSS, 2011).

A sequência gerada pelo algoritmo acima consiste de inteiros $X_n \in 0, 1, 2, ..., m-1$. A saída depende dos parâmetros m, a, c e da semente X_0. Se m, a e c são escolhidos de forma criteriosa, a sequência resultante se comporta como variáveis aleatórias uniformemente distribuídas (VOSS, 2011).

Outros métodos utilizados na geração de números pseudos aleatórios podem ser vistos no Anexo I, cujo conteúdo é uma reprodução do trabalho apresentado por Gomes (1994).

O Anexo I, citado em epígrafe, traz exemplos explanativos de Algoritmos Geradores de Números Aleatórios das seguintes distribuições de probabilidade:

1. Distribuição Exponencial;
2. Distribuição Normal;
3. Distribuição Beta;
4. Distribuição Gama;
5. Distribuição Binomial; e
6. Distribuição Poisson.

1.4.3 Qualidade dos Números Pseudo-Aleatórios

Segundo Voss (2011) números pseudo-aleatórios gerados em softwares como R-project e Matlab, são mais sofisticados, e apresentam maior complexidade que a do Algorítmo visto acima, ainda assim apresentam as seguintes características em comum:

1. depende da escolha de uma semente geradora, podendo se obter diferente sequências de números pseudo-aleatórios a partir da escolha de um valor X_a conhecido;
2. em todos os geradores as saídas são geralmente sequências periódicas, sendo a duração do período um indicador da qualidade de um gerador de números pseudo-aleatórios;
3. em praticamente todos os GNPA's os números gerados não são independentes, uma vez que, estes são gerados através de uma função determinística de um valor previamente escolhido; e
4. como normalmente são utilizados em substituição de valores i.i.d., é comum a realização de testes estatísticos que garantam a independência da sequência gerada.

Vale destacar que a qualidade dos números aleatórios está intimamente relacionada com a complexidade do algoritmo e o tamanho do período gerado. Ou seja, quanto maior o ciclo (período) maior a qualidade da sequência gerada.

1.4.4 Números Pseudo-aleatórios no Software R

O software livre R é um potente gerador de números Pseudo-aleatórios, sendo bastante utilizado para fins acadêmicos e para experimentos científicos.

O R utiliza as seguintes funções para geração de números pseudo-aleatórios:

O R utiliza diferentes técnicas para Gerar Números Aleatórios, armazenados em vetores chamados RNG's. Os usuários podem definir qual gerador utilizar (através do comando RNGkind) e qual a sua semente geradora (comando *set.seed*), podendo ser escolhido qualquer um dos geradores listados no Anexo II.

Função GNPA's no R

Tipo de Distribuição	**Comando**
Uniforme	runif
Normal	rnorm
t-Student	rt
Exponencial	rexp
Log Normal	rlnorm
Poisson	rpois
Beta	rbeta
Binomial	rbinom

Fonte: Elaborado pelo Autor

1.4.5 O Método de Monte Carlo

Segundo Voss (2011) são métodos computacionais, onde se examina propriedades de uma distribuição de probabilidade simulando uma grande amostra a partir de determinada distribuição e, em seguida estuda-se as propriedades estatísticas dessa amostra.

Tal método foi desenvolvido pelo Matemático John Von Neumann e formalizado em 1949, com a publicação do artigo "Monte Carlo Method", de John Von Neumann e Stanislaw Ulam.

O desenvolvimento se deu a partir dos avanços tecnológicos ocorridos após a segunda guerra mundial, que possibilitou o uso de sistemas de circuitos integrados para realizar operações matemáticas, e resolver equações em uma velocidade muito alta (RIHBANE, 2014).

> Segundo Metropolis e Ulam (1949 apud RIHBANE, 2014), a denominação Monte Carlo faz referência à famosa cidade de Mônaco, conhecida mundialmente como capital dos jogos de azar, pelo fato de Ulam e Von Neumann frequentarem e aplicarem matemática aos jogos de azar, observando aleatoriedade empírica. O método baseia-se em eventos que ocorrem de forma aleatória [...]

Segundo Voss (2011) o Método de Monte Carlo, é empregado para resolução de integrais complexas, a partir da equivalência existente entre o cálculo da esperança e o cálculo de integrais: Se $(X_i)_i \in N$ é uma sequência de i.i.d. variáveis aleatórias com a mesma distribuição como X, então

$$lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} f(X_i) = E\left(f\left(X\right)\right) \tag{1.12}$$

converge quase certamente. Enquanto a igualdade se mantiver no limite $N \to \infty$, nós podemos usar a aproximação

$$E\left(f\left(X\right)\right) \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i) \tag{1.13}$$

quando N é grande. Note que a estimativa do lado direito da equação é construído a partir de a valores aleatórios de X_i, que assim é uma quantidade aleatória, por si só.

O método de Monte Carlo é Baseado na Lei Forte dos Grandes Números e segundo Souza (2006 apud RIHBANE, 2014) consiste basicamente na geração de valores aleatórios com o objetivo de produzir "n" quantidades de cenários, em que a distribuição dos valores calculados deve refletir a probabilidade de ocorrências dos mesmos.

Para Nasser (2012) o método de Monte Carlo torna desnecessário escrever a equações diferenciais que descrevem o comportamento de sistemas complexos. A única exigência é que o sistema físico ou matemático seja modelado em termos de função de densidade de distribuição de probabilidade (FDP).

Depois de conhecida a FDP o procedimento consiste em fazer amostras aleatórias a partir da mesma, podendo o procedimento ser repetidos inúmeras vezes, sendo o resultado obtido através de técnicas estatísticas (descritivas e inferencial) (NASSER, 2012).

1.4.5.1 Em Inferência Estatística

Segundo Voss (2011) em Inferência Estatística o Método de Monte Carlo pode ser utilizado para cálculo do parâmetro T, então chamado de estimador, em especial para amostras com n pequeno em que a distribuição de T não é conhecida. Fazendo $T = \varphi(X_1, X_2, ..., X_n)$ uma função (mensurável) dos dados. Então $\mu = E\left(T\right)$ podendo ser estimada da seguinte forma:

1. gere uma amostra independente $(X_1^{(i)}, ..., X_n^{(i)})$ do modelo, com $i = 1, 2, ..., N$;
2. calcule $T^{(i)} = \varphi(X_1^{(i)}, ..., X_n^{(i)})$ para $i = 1, 2, ..., N$. Então $T^{(i)}$ são independentes e têm a mesma distribuição da estatística;
3. aproxime a esperança de T da seguinte forma

$$\mu \approx \hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} T^{(i)} \tag{1.14}$$

quando N é grande.

De maneira similar, a Função de Distribuição Acumulada $F(a) = P(T \leq a)$ pode ser estimada da seguinte forma:

$$F(a) \approx \hat{F}(a) = \frac{1}{N}\sum_{i=1}^{N} 1_{\left\{T^{(i)} \leq a\right\}} \tag{1.15}$$

Portanto, conforme Nasser (2012) o Método de Monte Carlo pode ser descrito como um método de simulação estatística que utiliza sequências de números aleatórios para desenvolver simulações. Ou seja, é um método universal utilizado para resolver problemas por meio de amostragem aleatória.

Segundo Nasser (2012), tal método é atualmente utilizado em diferentes campos de conhecimentos, dos quais citamos os seguintes:

a) **Atuária**: tábua de expectativa de vida, etc.;

b) **Finanças**: séries macroeconômicas, opções futuras, etc.;

c) **Computação gráfica**: redução de artefatos, espalhamento, etc;

d) **Geologia**: caracterização de reservatórios, etc.;

e) **Análise de Projetos**: opções reais;

f) **Jogo**: geração de redes (grafos).

1.4.6 Vantagens e Desvantagens de Simular

1.4.6.1 Vantagens

Para Gomes (1994) o uso da simulação Estatística apresenta algumas vantagens, as quais pode-se citar:

a. permite estimar a performance de um sistema existente, sob determinadas condições de operação;

b. permite a comparação de diferentes modelos alternativos de dado sistema (ou políticas alternativas de operação), de forma a descobrir aquele que mais satisfaz um determinado requisito;

c. é mais fácil controlar as condições experimentais, do que quando estamos experimentando com o próprio sistema;

d. permite estudar um sistema com evolução longa no tempo, como por exemplo um sistema econômico, em tempo reduzido, ou de forma alternativa, estudar trabalhos específicos, em tempo expandido.

1.4.6.2 Desvantagens

Ainda para Gomes (1994) a simulação estatística apresenta as seguintes desvantagens:

a. os métodos de simulação, em alguns casos, são dispendiosos e requerem muito tempo para a realização das suas diversas repetições;

b. a cada repetição, um modelo de simulação produz estimativas das verdadeiras características do modelo, para um conjunto particular de parâmetros de entrada. Portanto, várias repetições independentes do modelo serão eventualmente necessárias para uma correta interpretação estatística dos resultados;

c. o princípio de utilização é muito simples, contudo podem surgir dificuldades computacionais e no planejamento da experiência de amostragem. Outrossim, a interpretação do resultado de saída, pode não ser um exercício de estatística elementar.

1.4.7 Preâmbulo sobre R-Project e suas funções

Antes de iniciarmos a execução dos testes necessários para o alcance dos objetivos traçados para esse trabalho, cabe aqui, fazermos um preâmbulo sobre o o ambiente computacional a ser utilizado (o Software Livre R-Project), bem como, as duas das funções mais importantes utilizadas nos testes realizados a seguir.

1.4.7.1 R-Project

Segundo Peternelli e Mello (2011), o R é uma linguagem orientada a objetos criada em 1996 por Ross Ihaka e Robert Gentleman que permite a manipulação de dados, realização de cálculos e geração de gráficos. Semelhante à linguagem S desenvolvida pela AT&T's Bell Laboratories, mas com a vantagem de ser de livre distribuição.

> R fornece uma ampla variedade de estatística (linear e modelagem não-linear, testes estatísticos clássicos, análise de séries temporais, classificação, agrupamento,etc) e técnicas gráficas, e é altamente extensível. A linguagem S é muitas vezes o veículo de escolha para a investigação na metodologia estatística, e o R fornece uma rota de código aberto para a participação nessa atividade. Um dos pontos fortes do R é a facilidade com que gráficos bem delineados e de alta qualidade para impressão podem ser produzidos com possibilidade de inclusão de fórmulas e símbolos matemáticos quando necessário. (R Core Team, 2014).

Segundo Landeiro (2011) o R-Project, ou simplesmente R, é um software livre para computação estatística e construção de gráficos que pode ser baixado e distribuído gratuitamente de acordo com a licença GNU. O R está disponível para as plataformas UNIX, Windows e MacOS.

Ele se presta a diversas funções, desde uma calculadora científica, passando pela integração e derivação de funções matemáticas, até a realização de complexas análises estatísticas. (FERREIRA; OLIVEIRA, 2008)

Atualmente o R é uma importante ferramenta na análise e na manipulação de dados, com testes paramétricos e não paramétricos, modelagem linear e não linear, análise de séries temporais, análise de sobrevivência e simulação entre outros, além de ser possível elaborar diversos tipos de gráficos (PETERNELLI; MELLO, 2011).

1.4.7.2 Função Arima.sim

De acordo com o Manual disponibilizado no Athanasopoulos et al. (2014) é utilizada para simular modelos ARIMA.

A função pode ser usada da seguinte forma:
$arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, ...)n.start = NA, start.innov = rand.gen(n.start, ...), ...)$

Onde:

a. **Model:** é uma lista de componentes *ar* ou *ma* que são os coeficiente AR e MA, respectivamente. Opcionalmente, pode-se adicionar um componente relativo a ordem do modelo. Uma lista vazia gera um série temporal com modelagem ARIMA $(0, 0, 0)$, que é um ruído branco;

b. **n:** é o tamanho da série temporal, antes de ser diferenciada. É um inteiro estritamente positivo.

c. **rand.gen:** uma função que é chamada para gerar as inovações. Normalmente, rand.gen será um gerador de números aleatórios. Caso não seja definido será utilizada por padrão a função rnorm;

d. **n.start:** duração do período de arranque. Se NA, por padrão, um valor razoável é computado.

1.4.7.3 Função Auto.Arima

De acordo com informações disponibilizadas no Athanasopoulos et al. (2014) é uma função do pacote *Forecast* que retorna o melhor modelo ARIMA de acordo os critérios AIC, AICc ou BIC value. A função faz uma pesquisa sobre o melhor modelo possível (utilizando a técnica Stepwise), dentro das limitação da ordem, informadas pelo usuário.

O usuário pode entrar com o comando da seguinte forma:
$auto.arima(x, d = NA, D = NA, max.p = 5, max.q = 5, max.P = 2, max.Q = 2, max.order = 5, max.d = 2, max.D = 1, start.p = 2, start.q = 2, start.P = 1, start.Q = 1, stationary = FALSE, seasonal = TRUE)$

Onde:

a. **x:** é uma série temporal univariada;

b. **d:** é a ordem da primeira diferenciação. Se não informado será escolhido com base no teste KPSS;

c. **D:** é a ordem da diferenciação sazonal. Se não informado será escolhido com base no teste OCSB;

d. **Max.p, Max.d, Max.order:** são valores máximos dos respectivos argumentos;

e. **start.P, Star.Q, Star.D:** são valores iniciais dos respectivos argumentos;

f. **Stationary e Seasonal:** quando estes são **TRUE** restringem a busca aos respectivos modelos;

Estes são os argumentos mais comumente utilizados. Maiores detalhes sobre os demais argumentos podem ser encontrados na ajuda do Pacote *Forecast*, disponibilizado no (ATHANASOPOULOS et al., 2014).

As duas funções do R, vistas acima, executadas em lote, são amplamente utilizadas para gerar séries temporais simuladas assim como, no estudo dos componentes dos modelos das séries geradas.

Metodologia

2.1 Simulando Modelos AR(p) e MA(q)

Para análise da performance dos modelos simulados gerados pelo Software R, ulizaremos um conjunto de algoritmos, que irão gerar séries temporais simuladas, supostamente modeladas como AR(1) e AR(2) e MA(1) e MA(2). Para tanto será utilizada a função "arima.sim", "auto.arima", dentre outras, assim, como técnicas de análise exploratória e estatística descritiva (gráficos, correlograma, tabulação de dados, etc), para a análise da performance da função de simulação.

As séries simuladas os modelos AR(p) e MA(q), conforme expressões abaixo:
1. Modelo Autorregressivo

$$Z_t = \phi_1 Y_{t-1} + ... + \phi_p Y_{t-1} + \epsilon_t \tag{2.16}$$

2. Modelo de Médias Móveis

$$Z_t = \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - ... - \theta_q \epsilon_{t-q} \tag{2.17}$$

O conjunto de procedimentos, a ser executado pode ser resumido da seguinte forma:

1. definição do gerador de números pseudo-aleatórios e da semente geradora;
2. geração das n séries simuladas, de acordo com os argumentos passados para a função, bem como, e do modelo especificado;
3. detecção dos modelos das n séries simuladas, anteriormente geradas;
4. criação de tabela com as frequências absolutas dos modelos gerados, detectados anteriormente;
5. exibição de tabela com as frequências absolutas dos modelos gerados, bem como, gráfico comparativo entre o modelo proposto, e aqueles efetivamente gerados;
6. utilização da função para análise do uso das diferenças necessárias para tornar a série estacionária
7. correlogramas de 10 séries simuladas, amostradas de forma aleatória das séries anteriormente geradas.

Serão simuladas séries com os seguintes cenários:

1. para n pequeno ($n = 10, 20$) e n grande ($n = 100, 500$ *e* 1000);

2. variando-se a quantidade de repetições. O processo será repetido 10, 100 e 1000 vezes, gerando até 90.000 séries em cada execução do algorítmo;

3. alterando-se os modelos. Serão geradas séries simuladas para processos autorregressivos (AR) e de Médias Móveis (MA);

4. alterando-se os valores dos parâmetros (p) e (q). Serão simuladas séries para modelos $AR(1)$ e $AR(2)$, $MA(1)$ e $MA(2)$;

5. alterando-se os coeficientes (θ) e (ϕ). Serão simuladas séries para modelos com coeficientes variando de 0.10 a 0.90, havendo um incremento de 0.10 para cada repetição;

As séries geradas nos cenários acima, serão analisadas e através da comparação dos resultados, poderemos responder aos seguintes questionamentos:

a. será que a performance da função *arima.sim* é afetada pelo número de observações (para n pequeno ou n grande)?;

b. qual a percentagem de séries temporais simuladas, corresponde ao modelo teórico escolhido pelo usuário?

c. a função de simulação do R é mais precisa em simuar modelos Autorregressivos ou modelos de Médias Móveis?

d. a performance da função de simulação (*arima.sim*) é afetada pela escolha dos valores dos parâmetros (p) ou (q) dos modelos AR ou MA?

e. diferentes valores para os coeficientes utilizados influenciam na performance da função arima.sim?

2.1.1 Execução

Visando responder aos questionamentos acima executamos o algoritmo em questão, e registramos os resultados das simulações, da seguinte forma:

1) simulamos séries temporais para modelos Autorregressivos de Ordem 1 $AR(1)$, utilizando o GNPA *"Mersenne-Twister"* com o valor do parâmetro ϕ_1 variando de 0.10 *a* 0.90 (sendo incrementado 0.10 a cada *"loop"* do algorítmo) gerando assim, séries ajustadas para 9 diferentes valores dos parâmetros;

2) o algoritmo incialmente simula séries de tamanho $n = 10$ observações e 10 simulações. Pequenas modificações são realizadas no código inicial e então, podemos observar séries de tamanho maiores ($n = 20, n = 100, n = 500$ *e* $n = 1000$) variando, posteriormente, o número de séries simuladas ($10, 100$ *e* 1000);

3) a posteriori, e ainda utilizando o "Mersenne-Twister" como GNPA, foram simulados modelos Autorregressivos de Ordem 2 $AR(2)$, com a mesma variação para o parâmetro ϕ_2 e com $\phi_1 = 0.01$. Respeitando-se dessa forma os requisitos de estacionariedade dos modelos gerados;

4) procedimentos análogos foram utilizados na geração de séries temporais simuladas dos Modelos de Médias Móveis de Ordem 1 $MA(1)$ e de Ordem 2 $MA(2)$. Devemos ressaltar que agora os parâmetros da série são θ_1 e θ_2.

Portanto, podemos resumir as simulações realizadas da seguinte forma:

Tabela 1 – Resumo das Simulações

Modelo	**Nº Observações**	**Nº Simulações**	**Parâmetros**	**GNPA**
AR(1)	10, 20, 100, 500 e 1000	10, 100 e 1000	ϕ_1: de 0.10 a 0.90	Mersenne-Twister
AR(2)	10, 20, 100, 500 e 1000	10, 100 e 1000	ϕ_1:0.01 e ϕ_2: de 0.10 a 0.90	Mersenne-Twister
MA(1)	10, 20, 100, 500 e 1000	10, 100 e 1000	θ_1: de 0.10 a 0.90	Mersenne-Twister
MA(2)	10, 20, 100, 500 e 1000	10, 100 e 1000	θ_1:0.01 e θ_2: de 0.10 a 0.90	Mersenne-Twister

Fonte: Elaborado pelo autor

Pelos dados informados acima, podemos observar que foram registrados informações de 60 diferentes cenários, podendo então ser verificada a performance da função *arima.sim* na simulação dos modelos teóricos acima propostos.

Note que para 10 simulações o algoritmo gera 90 séries diferentes; para 100 simulações são geradas 900 séries e, finalmente, para 1000 simulações 9000 séries diferentes são geradas pelo comando *arima.sim* e analisadas pela função *auto.arima*.

Classificaremos o desempenho da função do R, *arima.sim*, de acordo com as frequências dos modelos teóricos gerados, conforme critério abaixo:

frequência entre 0% e 50% = $Fraco$ (F)

frequência entre 51% e 69% = Bom (B)

frequência acima de 69% = Ó$timo$ (O)

Resultados e Discussões

Vamos analisar os resultados, dividindo os cenários das simulações em dois grandes subgrupos, utilizando como critério de classificação os modelos teóricos propostos: Autorregressivos e de Médias Móveis.

3.1 Modelos Autorregressivos

AR(1): gerando 10 séries simuladas com apenas 10 observações podemos observar que para $\phi_1 = 0.2, 0.3, 0.8$ *e* 0.5 não foram geradas séries com a ordem teórica proposta $AR(1)$ e sim, séries $AR(0)$ para os demais valores de ϕ_1, apesar da série gerada pertencer ao modelo teórico proposto, a frequência gira em torno de 10% e 20%. Segue abaixo as tabelas de frequencia das simulações geradas:

Tabela 2 – Simulação AR(1), n=10

Valores de ϕ_1	Ordem	
	0	**1**
0.1	8	2
0.2	10	0
0.3	10	0
0.4	9	1
0.5	10	0
0.6	9	1
0.7	8	2
0.8	10	0
0.9	9	1

Tabela 3 – Simulação AR(1), n=20

Valores de ϕ_1	Ordem		
	0	**1**	**2**
0.1	7	1	2
0.2	9	0	1
0.3	9	1	0
0.4	7	3	0
0.5	4	6	0
0.6	6	3	1
0.7	8	2	0
0.8	9	1	0
0.9	4	5	1

Tabela 4 – Simulação AR(1), n=100

Valores de ϕ_1	Ordem		
	0	**1**	**2**
0.1	8	2	0
0.2	7	2	1
0.3	7	3	0
0.4	6	4	0
0.5	5	5	0
0.6	3	6	1
0.7	5	4	1
0.8	1	7	2
0.9	5	5	0

Tabela 5 – Simulação AR(1), n=1000

Valores de ϕ_1	Ordem				
	0	**1**	**2**	**3**	**4**
0.1	6	3	0	0	1
0.2	6	4	0	0	0
0.3	2	4	2	2	0
0.4	0	7	2	1	0
0.5	0	10	0	0	0
0.6	0	8	0	0	0
0.7	0	5	3	2	0
0.8	0	6	2	2	0
0.9	1	6	2	1	0

Fonte: Elaborado pelo autor

Abaixo, podemos observar os gráficos, calculados a partir das frequências dos modelos gerados através da função *arima.sim*, com $n = 10$, $n = 20$, $n = 100$ e $n = 1000$:

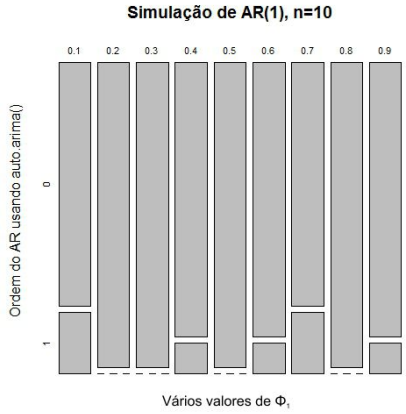

Figura 11 – (a) - AR(1)

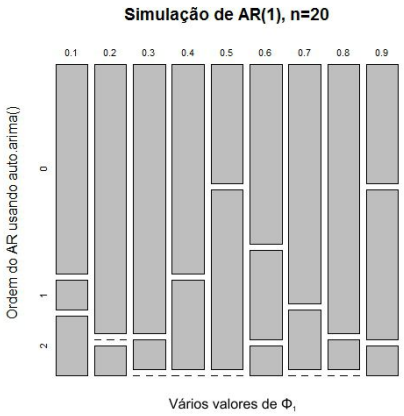

Figura 12 – (b) - AR(1)

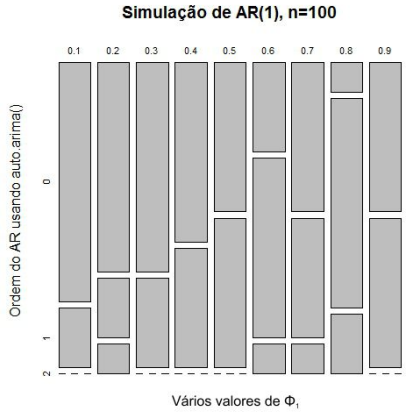

Figura 13 – (c) - AR(1)

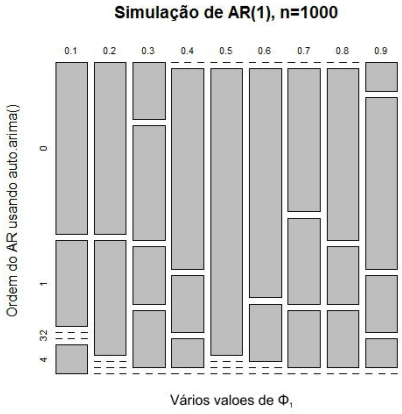

Figura 14 – (d) - AR(1)

Fonte: Elaborado pelo autor

Note que a performance varia de acordo com os valores de ϕ_1, bem como, com a variação dos valores de n.

Podemos então, a partir dos critérios definidos anteriormente, classificar a performance da função *arima.sim* para o cenário em questão, da seguinte forma:

Tabela 6 – Modelos AR(1)

Com 10 Simulações					
ϕ_1	10	20	100	500	1000
0.1	F	F	F	F	F
0.2	F	F	F	F	F
0.3	F	F	F	O	F
0.4	F	F	F	B	O
0.5	F	F	F	O	O
0.6	F	F	B	O	O
0.7	F	F	F	F	F
0.8	F	F	O	O	B
0.9	F	F	F	F	B

Fonte: Elaborado pelo autor

Destaque para a baixíssima performance apresentada, pela função *arima.sim* na geração do modelo teórico proposto ($AR(1)$) para séries com valores pequenos ($n < 30$).

Aumentamos então o número de simulações com a finalidade de verificarmos se há uma melhora na perfomance dos modelos gerados.

A seguir exibimos os gráficos gerados para as frequências de 100 simulações do modelo $AR(1)$ com o parâmetro $\phi_1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ e $n = (10, 20, 100, 500 \text{ e } 1000)$:

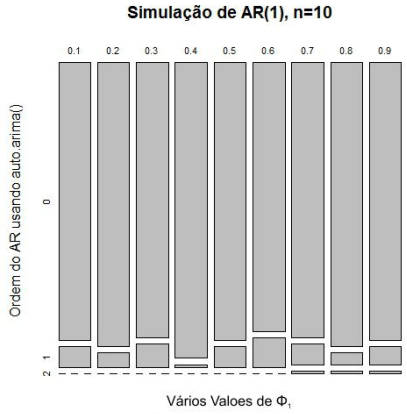

Figura 15 – (a) - AR(1)

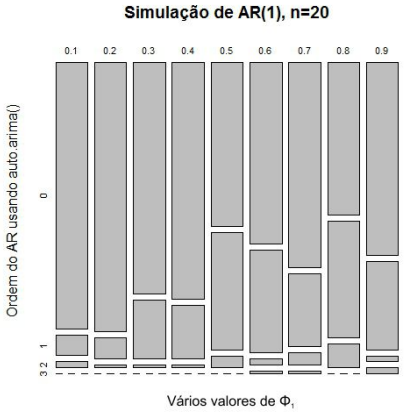

Figura 16 – (b) - AR(1)

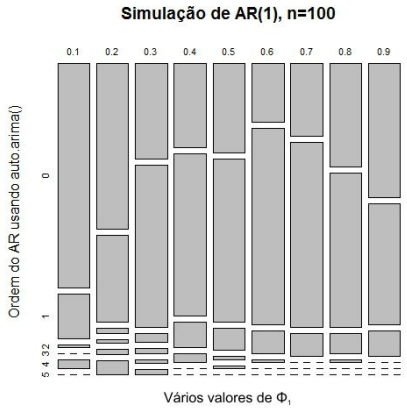

Figura 17 – (c) - AR(1)

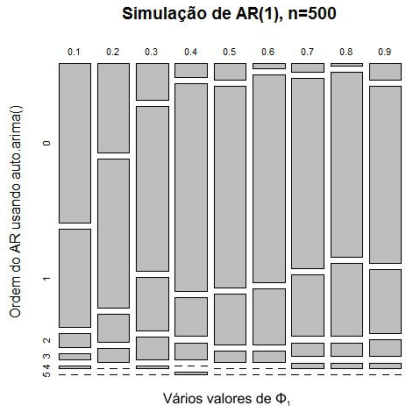

Figura 18 – (d) - AR(1)

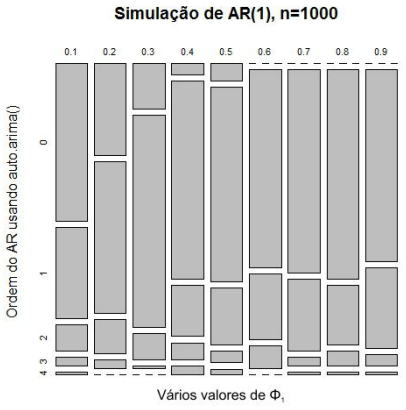

Figura 19 – (e) - AR(1)

Fonte: Elaborado pelo autor

Consolidando as informações das tabelas de frequências dos gráficos acima demonstrados, podemos verificar a seguinte performance da função *arima.sim* na geração de 100 séries temporais com o parâmetro $\phi_1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ e $n = (10, 20, 100, 500\ e\ 1000)$:

Tabela 7 – Modelos AR(1)

Com 100 Simulações					
ϕ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	F
0.2	F	F	F	B	B
0.3	F	F	B	B	O
0.4	F	F	B	O	B
0.5	F	B	B	O	B
0.6	F	F	O	O	B
0.7	F	F	B	B	O
0.8	F	F	B	B	O
0.9	F	F	F	B	B

Fonte: Elaborado pelo autor

Note que a partir de $n = 100$ observações, houve um aumento significativo na frequência em que a série gerada corresponde a ordem do modelo teórico proposto. Destaque para $n = 1000$ e ϕ_1 de 0.2 a 0.9 cuja performance das simulações varia entre Boa e Ótima.

Contudo, podemos verificar que o aumento do número de simulações, no que concerne as séries temporais com n pequeno, não altera a performance da função *arima.sim*, continuando fraca, para todos os valores de ϕ_1.

Analisaremos então a performance dos modelos simulados $AR(1)$ gerando 1000 séries para o parâmetro $\phi_1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ e $n = (10, 20, 100, 500\ e\ 1000)$:

As tabelas de frequências geradas foram:

Tabela 8 – Simulação AR(1), n=10

Valores de ϕ_1	Ordem		
	0	**1**	**2**
0.1	952	45	3
0.2	963	34	3
0.3	942	53	5
0.4	938	56	6
0.5	936	60	4
0.6	908	86	6
0.7	915	76	9
0.8	918	74	8
0.9	926	62	12

Tabela 9 – Simulação AR(1), n=20

Valores de ϕ_1	Ordem				
	0	**1**	**2**	**3**	**4**
0.1	894	86	18	2	0
0.2	871	109	18	1	1
0.3	812	162	23	3	0
0.4	770	214	15	1	0
0.5	685	297	17	1	0
0.6	606	360	31	3	0
0.7	569	389	41	1	0
0.8	556	407	33	4	0
0.9	607	349	41	3	0

Fonte: Elaborado pelo autor

Tabela 10 – Simulação AR(1), n=100

Valores de ϕ_1	Ordem					
	0	**1**	**2**	**3**	**4**	**5**
0.1	770	164	28	13	13	12
0.2	568	363	37	19	7	6
0.3	430	492	44	15	8	11
0.4	325	589	55	17	7	7
0.5	279	651	60	4	5	1
0.6	239	665	91	4	1	0
0.7	278	631	83	7	1	0
0.8	359	555	78	8	0	0
0.9	569	367	60	3	1	0

Tabela 11 – Simulação AR(1), n=500

Valores de ϕ_1	Ordem					
	0	**1**	**2**	**3**	**4**	**5**
0.1	576	341	68	13	2	0
0.2	386	484	103	22	4	1
0.3	220	604	145	24	6	1
0.4	94	709	159	28	8	2
0.5	62	716	174	38	8	2
0.6	29	744	179	38	7	3
0.7	12	707	238	38	5	0
0.8	128	586	226	45	12	3
0.9	569	367	60	3	1	0

Tabela 12 – Simulação AR(1), n=1000

Valores de ϕ_1	Ordem					
	0	**1**	**2**	**3**	**4**	**5**
0.1	541	362	80	12	5	0
0.2	328	517	125	22	8	0
0.3	123	691	147	32	5	2
0.4	66	710	181	35	7	1
0.5	26	713	212	36	8	5
0.6	14	727	193	51	12	3
0.7	5	713	236	37	9	0
0.8	5	695	230	59	10	1
0.9	26	670	238	51	10	5

Fonte: Elaborado pelo autor

Usando o mesmo critério de classificação citado anteriormente, podemos definir, a partir da análise das tabelas de frequência acima, o desempenho da função na geração de 1000 séries para o parâmetro $\phi_1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ e $n = (10, 20, 100, 500\ e\ 1000)$ conforme apresentado na Tabela 3.

Note que o desempenho apresentado para esse cenário é muito parecido com aquele estudado anteriormente ($AR(1)$ com 100 simulações), havendo uma aparente melhoria nas séries simuladas para n grande (> 30).

Tabela 13 – Modelos AR(1)

Com 1000 Simulações					
ϕ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	F
0.2	F	F	F	F	B
0.3	F	F	F	B	O
0.4	F	F	B	O	O
0.5	F	B	B	O	O
0.6	F	F	B	O	O
0.7	F	F	B	O	O
0.8	F	F	B	O	O
0.9	F	F	F	B	B

Fonte: Elaborado pelo autor

AR(2): passaremos então a analisar a performance da função *arima.sim* na simulação de modelos $AR(2)$:

O modelo $AR(2)$ foi simulado variando-se o valor de ϕ_2, assim como no modelo $AR(1)$, ou seja $\phi_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ e, com o valor de $\phi_1 = 0.01$.

Para o modelo em questão e, analisando um cenário com 10 simulações, observamos que apenas a partir de $n = 100$ temos resultados significativos, no tocante a geração de modelos correspondentes ao modelo teórico proposto. De um modo geral, podemos afirmar que gerando 10 simulações os resultados obtidos são semelhantes aos das séries $AR(1)$, ou seja, uma performance **Fraca** na geração de séries com n pequeno e uma significativa melhoria ocorrida com o aumento do número de observações simuladas (n grande).

Segue abaixo tabela com avaliação da performance da função *arima.sim* em relação à simulação de modelos $AR(2)$, para 10 simulações:

Tabela 14 – Modelos AR(2)

Com 10 Simulações					
ϕ_2	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	F
0.2	F	F	F	F	F
0.3	F	F	F	F	B
0.4	F	F	F	B	B
0.5	F	B	F	O	O
0.6	F	F	F	F	O
0.7	F	F	F	B	F
0.8	F	F	F	B	F
0.9	F	F	B	F	B

Fonte: Elaborado pelo autor

Na página seguinte são exibidos os gráficos gerados a partir das frequências dos modelos das 100 séries temporais simuladas para cada valor do parâmetro $\phi_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$:

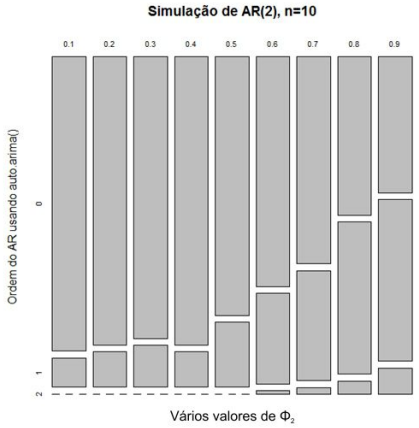

Figura 20 – (a) - AR(2)

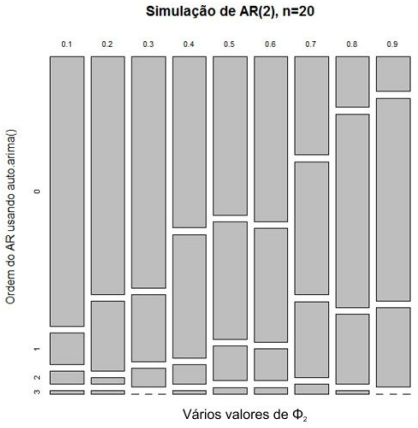

Figura 21 – (b) - AR(2)

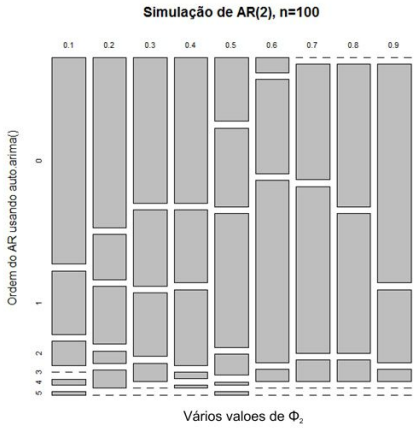

Figura 22 – (c)- AR(2)

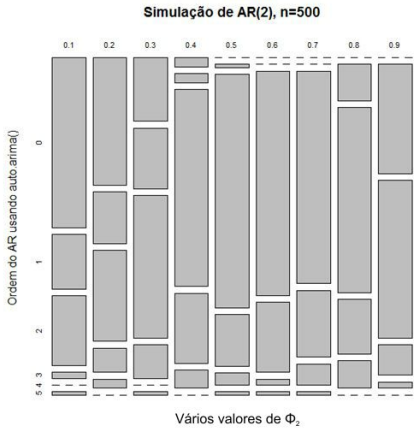

Figura 23 – (d)- AR(2)

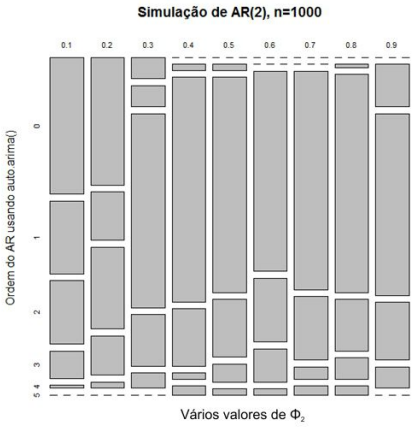

Figura 24 – (e)-AR(2)

Fonte: Elaborado pelo autor

Comparando os gráficos acima com aqueles gerados a partir das fequências dos modelos gerados nas séries simuladas com o Modelo Teórico ($AR(1)$), podemos verificar

que visualmente apresentam diferenças significativas. Contudo, tal impressão não resiste a uma análise mais detida dos mesmos. Podemos então notar claramente que para a maioria dos valores de ϕ_2 a quantidade de simulações que são condizentes com a ordem teórica proposta, não ultrapassam 50% e assim, como acontecia nas simulações dos modelos $AR(1)$, tal fato, só se altera para as amostras simuladas com $n \geq 100$.

Tabela 15 – Modelos AR(2)

Com 100 Simulações					
ϕ_2	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	F
0.2	F	F	F	F	F
0.3	F	F	F	F	B
0.4	F	F	F	B	O
0.5	F	B	F	O	O
0.6	F	F	B	O	B
0.7	F	F	B	O	O
0.8	F	F	F	B	O
0.9	F	F	F	B	B

Fonte: Elaborado pelo autor

Assim como nos gráficos, a comparação com a tabela correspondente, gerada no estudo do modelo $AR(1)$, revela que a diferença de performance da função na geração do modelo proposto ($AR(2)$), não sofre grandes alterações.

Gerando 1000 simulações para cada valor do parâmetro $\phi_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ $\phi_1 = 0.01$, os gráficos gerados a partir das frequências dos modelos simulados, possuem características semelhantes aos anteriormente estudados e, devido a esse fato não serão aqui exibidos. Ao invés disso, mostraremos a seguir a tabela da avaliação da performance das séries temporais simuladas ($AR(2)$ com 1000 repetições):

Tabela 16 – Modelos AR(2)

Com 1000 Simulações					
ϕ_2	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	F
0.2	F	F	F	F	F
0.3	F	F	F	F	B
0.4	F	F	F	B	O
0.5	F	F	F	O	O
0.6	F	F	B	O	O
0.7	F	F	B	O	O
0.8	F	F	F	B	B
0.9	F	F	F	F	B

Fonte: Elaborado pelo autor

3.2 Modelos de Médias Móveis

Agora, para a avaliação da performance da função do R **arima.sim** na simulação de modelos de Modelos de Médias Móveis, utilizamos todos os procedimentos aplicados aos modelos Autorregressivos, vistos acima. Ou seja, foram simulados cenários com 10, 100 e 1000 repetições, simulando séries inicialmente de Médias Móveis de Ordem 1 *MA(1)* e a posteriori de Ordem 2 *MA(2)*, variando-se os valores do parâmetro θ_1 (para *MA(1)*) e os valores de θ_2 (para *MA(2)*).

MA(1): Para um cenário com 10 simulações foram observadas as seguintes frequências:

Tabela 17 – Simulação MA(1), n=10

Valores de θ_1	Ordem	
	0	**1**
0.1	9	1
0.2	10	0
0.3	10	0
50.4	10	0
0.5	10	0
0.6	9	1
0.7	9	1
0.8	9	1
0.9	10	0

Tabela 18 – Simulação MA(1), n=20

Valores de θ_1	Ordem		
	0	**1**	**2**
0.1	8	1	1
0.2	9	1	0
0.3	9	1	0
0.4	6	4	0
0.5	5	5	0
0.6	7	3	0
0.7	6	3	1
0.8	5	4	1
0.9	4	6	0

Tabela 19 – Simulação MA(1), n=100

Valores de θ_1	Ordem		
	0	**1**	**2**
0.1	9	1	0
0.2	7	3	0
0.3	5	5	0
0.4	2	8	0
0.5	3	5	2
0.6	0	10	0
0.7	0	10	0
0.8	0	10	0
0.9	1	9	0

Tabela 20 – Simulação MA(1), n=500

Valores de θ_1	Ordem		
	0	**1**	**2**
0.1	5	5	0
0.2	3	6	1
0.3	3	6	1
0.4	0	8	2
0.5	1	9	0
0.6	0	7	3
0.7	0	8	2
0.8	0	9	1
0.9	0	10	0

Fonte: Elaborado pelo autor

Tabela 21 – Simulação MA(1), n=1000

Valores de θ_1	Ordem				
	0	**1**	**2**	**3**	**4**
0.1	4	6	0	0	0
0.2	3	7	0	0	0
0.3	1	7	2	0	0
0.4	0	10	0	0	0
0.5	0	8	2	0	0
0.6	0	8	2	0	0
0.7	0	3	4	2	1
0.8	0	9	1	0	0
0.9	0	8	1	1	0

Fonte: Elaborado pelo autor

A avaliação da performance das simulações, seguindo os mesmos critérios definidos para os Modelos Autorregressivos foi a seguinte:

Tabela 22 – Modelos MA(1)

Com 10 Simulações					
θ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	B
0.2	F	F	F	B	O
0.3	F	F	F	B	O
0.4	F	F	O	O	O
0.5	F	F	F	O	O
0.6	F	F	O	O	O
0.7	F	F	O	O	F
0.8	F	F	O	O	O
0.9	F	F	O	O	O

Fonte: Elaborado pelo autor

Comparando a tabela acima com a tabela 6, referente à avaliação da performance da função *arima.sim* na geração de modelos *AR(1)* (com 10 simulações) podemos verificar uma melhora acentuada na simulação de séries temporais *MA(1)*, principalmente para os valores de $\theta_1 > 0.5$ e, com $n \geq 100$. Pode-se observar na tabela 19, que para $n = 100$ e os valores de $\theta = (0.6, 0.7\ e\ 0.8)$, 100% das séries geradas correspondem à ordem do modelo teórico proposto.

Vejamos se tal fato se confirma, avaliando a performance da função na simulação de 100 séries temporais.

Os resultados podem ser vistos na Tabela 23, logo abaixo.

Tabela 23 – Modelos MA(1)

Com 100 Simulações					
θ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	B
0.2	F	F	F	B	B
0.3	F	F	B	O	O
0.4	F	F	O	O	O
0.5	F	F	O	O	O
0.6	F	F	O	O	O
0.7	F	B	O	O	O
0.8	F	B	O	O	O
0.9	F	F	O	O	O

Fonte: Elaborado pelo autor

Note que há uma ligeira melhora na performance da geração das séries simuladas. Um maior número de simulações, está em consonância com o modelo teórico proposto. Note ainda, que mesmo para $n = 20$, já pode-se observar frequências de modelos corretamente simulados, com quantitativo maior que 50% (para $\theta = 0.8$ *e* 0.9).

Agora passaremos a analisar a performance da função **arima.sim** na geração de 1000 simulações de Séries *MA(1)* para os valores do parâmetro $\theta_1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$. A análise da frequência do modelo teórico gerado, resultou na seguinte avaliação:

Tabela 24 – Modelos MA(1)

Com 1000 Simulações					
θ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	B	B
0.2	F	F	F	B	B
0.3	F	F	B	O	O
0.4	F	F	O	O	O
0.5	F	F	O	O	O
0.6	F	F	O	O	O
0.7	F	B	O	O	O
0.8	F	B	O	O	O
0.9	F	F	O	O	O

Fonte: Elaborado pelo autor

Comparando a Tabela 23 e a Tabela 24, podemos verificar que a performance da função *arima.sim* na geração de séries *MA(1)*, permanece praticamente inalterada. Ou seja, são simuladas séries, que na sua maioria corresponde ao modelo teórico proposto. Ressaltamos que, as frequências relativas são praticamente idênticas, tanto para 100 simulações quanto para 1000 simulações.

Na página seguinte vemos os gráficos gerados para o cenário descrito acima (1000 simulações e $\theta_1 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$

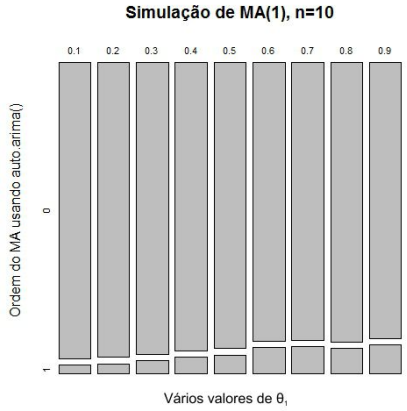

Figura 25 – (a) - MA(1)

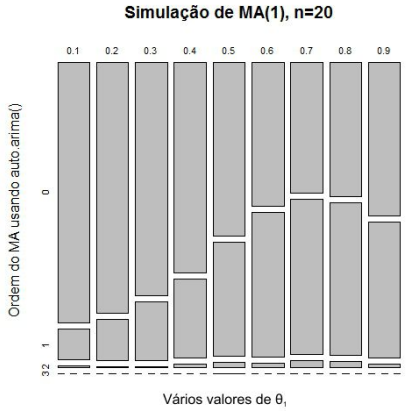

Figura 26 – (b) - MA(1)

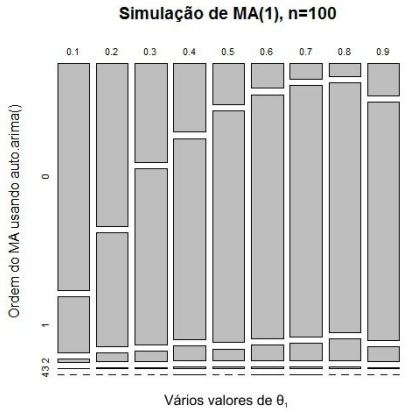

Figura 27 – (c) - MA(1)

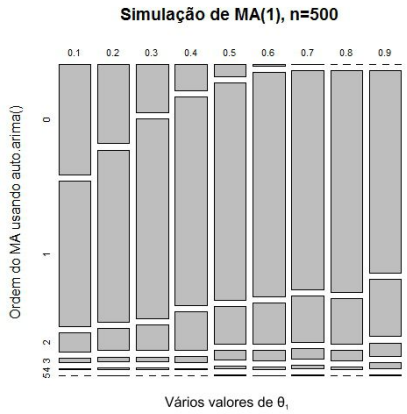

Figura 28 – (d) - MA(1)

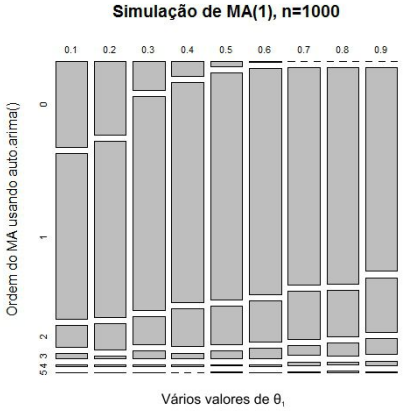

Figura 29 – (e) - MA(1)

Fonte: Elaborado pelo autor

Os gráficos acima, sugerem que para valores de θ_1 mais próximos de 1 nos fornece simulações mais precisas, ou seja, mais frequentemente as séries simuladas, guardam

consonância com a ordem do modelo originalmente proposto, assim como, podemos ainda observar que aumentando o valor de n a performance das simulações também aumenta de forma significativa.

MA(2): passaremos então a analisar a performance da função *arima.sim* na simulação de modelos $MA(2)$:

Vale ressaltar que assim como o modelo Autorregressivo de ordem 2 ($AR(2)$), para o modelo *MA(2)*, serão simuladas séries variando os valores de θ_2 de (0.1 a 0.9) e $\theta_1 = 0.01$.

Mantendo-se o cenário anteriormente especificado para 10 simulações, podemos verificar a seguinte performance, da função *arima.sim*:

Tabela 25 – Modelos MA(2)

Com 10 Simulações				
θ_1	n=10	n=20	n=100	n=500
0.1	F	F	F	F
0.2	F	F	B	B
0.3	F	F	F	O
0.4	F	F	F	O
0.5	F	F	F	O
0.6	F	F	F	O
0.7	F	F	F	B
0.8	F	F	F	O
0.9	F	F	F	O

Fonte: Elaborado pelo autor

Assim como os demais modelos estudados, a simulação de séries temporais para modelos *MA(2)* com n pequeno teve um desempenho fraco. Contudo devemos ressaltar que, nesse experimento as 10 simulações geradas para os valores de $n = 10$ e $n = 20$, 100% destas não corresponderam ao modelo originalmente proposto. Diferentemente das simulações anteriores, apenas para $n = 500$, ocorreu um desempenho satisfatório na geração das séries.

Ressaltamos ainda que, para $n = 1000$ foram geradas mensagens de erro no R *warnings()*, relatando quanto da impossibilidade de ajuste final do modelo utilizando máxima verossimilhança. Tal mensagem havia ocorrido nos demais experimentos, contudo não pareceu afetar tanto o resultado final das simulações.

Com o intuito de melhor avaliar as séries simuladas, a cada experimento sorteamos aleatoriamente através da função $runif$ (com semente geradora 40) um amostra de tamanho 10 e testamos tais séries, fazendo uso da função *ndiffs* (que serve para verificar a quantidade de diferenciações que uma série deveria sofrer para torna-se estacionária (ATHANASOPOULOS et al., 2014)) conjuntamente com a função *acf* (que calcula e exibe o correlograma da série temporal (ATHANASOPOULOS et al., 2014)).

Para esse experimento, notamos que os correlogramas exibidos, assim como, os valores calculados pelas funções *ndiff* e *auto.arima* sugerem que muitas das séries geradas são passeios aleatórios, ou seja, ARIMA(0,0,0).

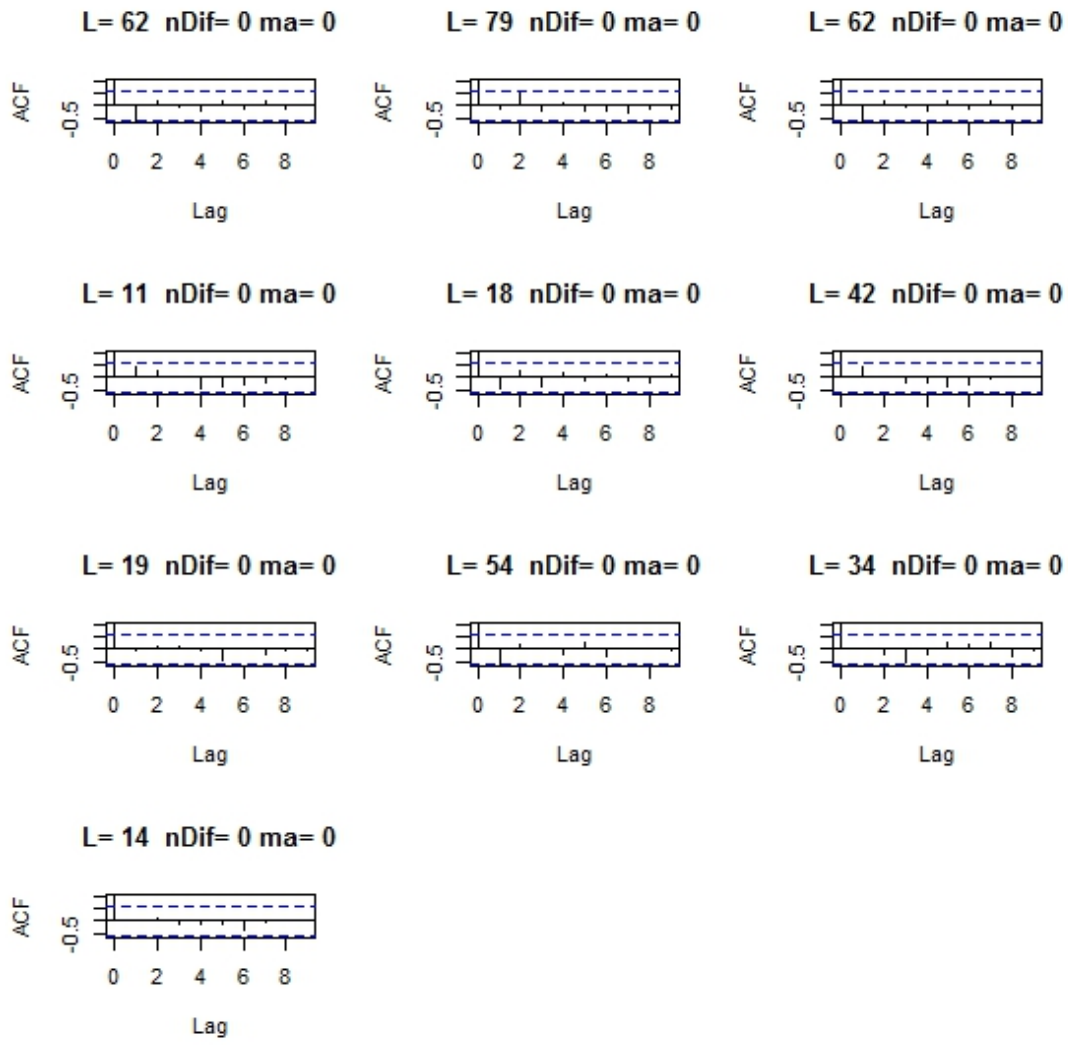

Figura 30 – Correlograma - $n = 10$

Fonte: Elaborado pelo autor

Agora, gerando 100 simulações *MA(2)* obtivemos os seguintes resultados:

Tabela 26 – Modelos MA(2)

Com 1000 Simulações					
θ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	B
0.2	F	F	F	O	O
0.3	F	F	B	O	O
0.4	F	F	B	O	O
0.5	F	F	O	O	O
0.6	F	F	O	O	O
0.7	F	F	O	B	O
0.8	F	F	O	O	O
0.9	F	F	F	B	B

Fonte: Elaborado pelo autor

Para 100 simulações, e mantendo-se os demais parâmetros inalterados, podemos notar que o desempenho da função *arima.sim* na simulação de séries $MA(2)$, teve desempenho correlato com o que vimos nos modelos anteriormente estudados. Podemos notar nos gráficos abaixo, que assim como acontecia em $MA(1)$, aumentando-se o valor de n e os valores do parâmetro θ_2, aumenta também a frequência que o modelo gerado é o mesmo que aquele originalmente proposto.

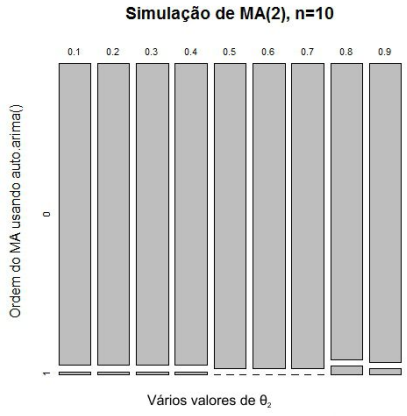

Figura 31 – (a) - MA(2)

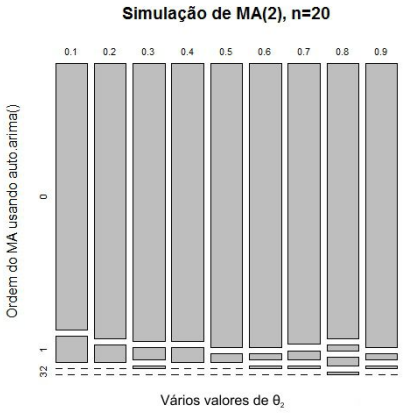

Figura 32 – (b)- MA(2)

Fonte: Elaborado pelo autor

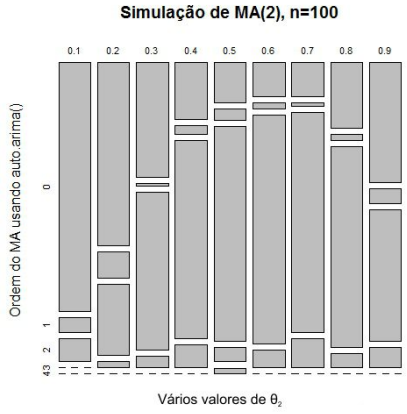

Figura 33 – (c) - MA(2)

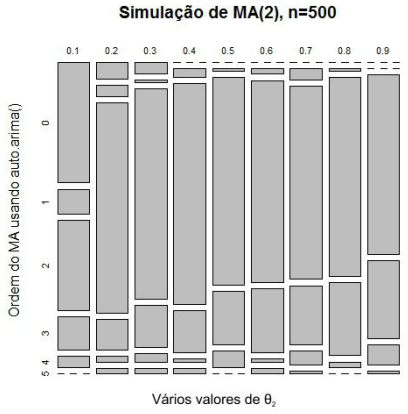

Figura 34 – (d)- MA(2)

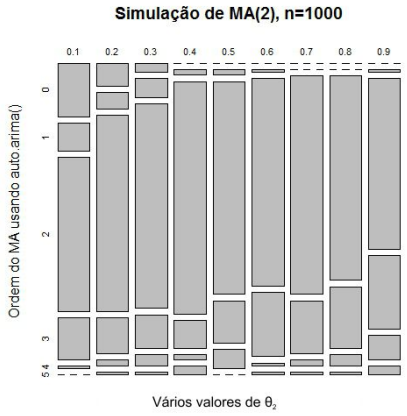

Figura 35 – (e)- MA(2)

Fonte: Elaborado pelo autor

Cabe aqui fazer uma pausa para falarmos sobre os resultados apresentados na Tabela 25, que avalia a performance dos modelos simulados de apenas 10 séries *MA(2)*. Os resultados apresentados divergem do que analisamos até então, sugerindo ter havido alguma falha de ordem técnica, ou algum problema relacionado a semente geradora escolhida. O fato é que para descobrirmos tal falha necessitaríamos analisar de forma mais detalhada, apenas este caso específico, fugindo da proposta inicial deste trabalho.

Feito essa observação, podemos por fim, analisar a simulação de 1000 séries temporais, para o modelo *MA(2)*, variando os valores do parâmetro $\theta_2 = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ e $\theta_1 = 0.01$.

A performance da função do R, *arima.sim* foi registrada na tabela abaixo:

Tabela 27 – Modelos MA(2)

Com 100 Simulações					
θ_1	n=10	n=20	n=100	n=500	n=1000
0.1	F	F	F	F	B
0.2	F	F	F	O	O
0.3	F	F	F	O	O
0.4	F	F	B	O	O
0.5	F	F	O	O	O
0.6	F	F	O	O	O
0.7	F	F	O	B	O
0.8	F	F	O	B	B
0.9	F	F	O	B	B

Fonte: Elaborado pelo autor

E, assim como ocorreu para as 100 simulações do modelo *MA(2)*, mantendo-se os demais parâmetros inalterados, aumentando-se o valor de n e para os valores θ_2 mais próximos de 1, a frequência em que a série gerada corresponde ao modelo inicialmente proposto, também aumenta, resultando numa melhor performance das simulações.

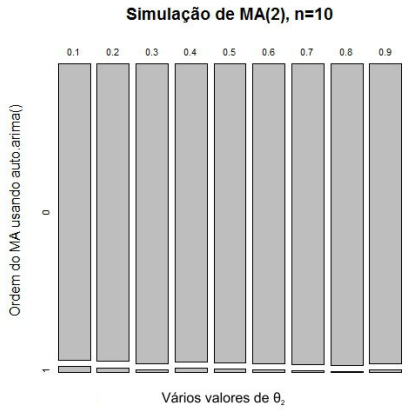

Figura 36 – (a) - MA(2)

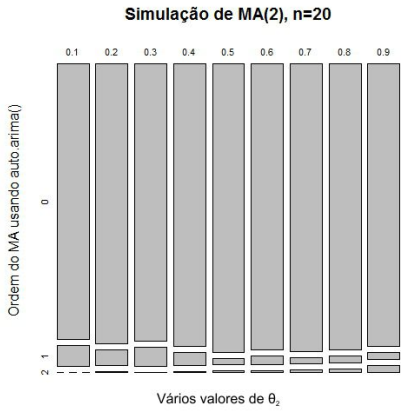

Figura 37 – (b)- MA(2)

Fonte: Elaborado pelo autor

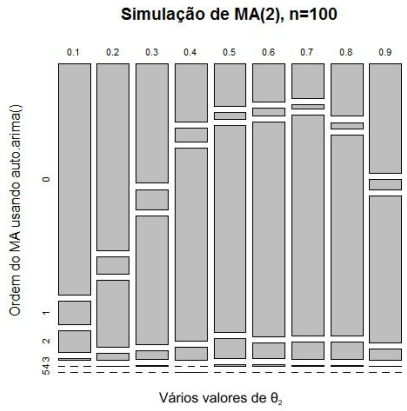

Figura 38 – (c)- MA(2)

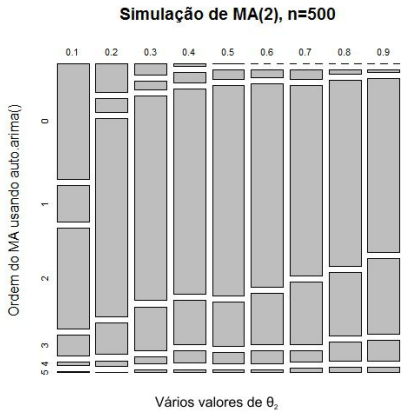

Figura 39 – (d)- MA(2)

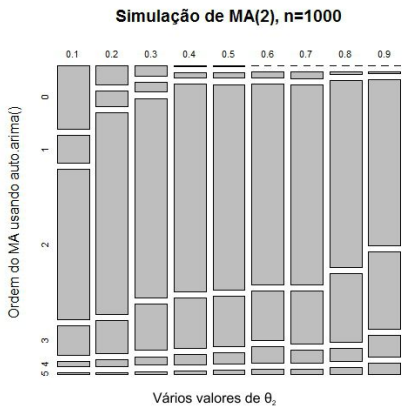

Figura 40 – (e)- MA(2)

Fonte: Elaborado pelo autor

Conclusão

Podemos concluir então que embora o Software R seja, atualmente, bastante utilizado no meio acadêmico-científico para geração de dados simulados, especialmente nos casos em que não é possível termos acesso aos dados reais, seu uso, na geração de séries temporais com modelos *AR(1), AR(2), MA(1) e MA(2)* deve ser visto com certa cautela.

Conforme podemos observar no decorrer dos experimentos, a frequência em que o Modelo gerado corresponde àquele originalmente proposto, nem sempre é satisfatória. Em muitos casos, especialmente para séries simuladas com poucas observações, ou seja, $n \leq 20$, cem por cento dos modelos gerados não correspondiam ao modelo simulado, necessitando de um elevado número de simulações para que fossem geradas séries com os modelos originalmente especificados. Poucos foram os experimentos em que 100% das simulações correspondiam ao modelo especificado, podendo aqui ser citados 10 simulações de modelos *MA(1)*.

Ressaltamos que a partir da observação atenta dos correlogramas das séries geradas, em especial os calculados para o modelo *MA(2)* (com 10 simulações), podemos observar que muitas das simulações geradas são passeios aleatórios, tratando-se portanto, de processos não estacionários, fugindo completamente às especificações originalmente propostas.

Podemos ainda concluir que elevando-se a quantidade de simulações, e a partir de 100 observações, normalmente, mais de 50% das séries simuladas apresentam consonância com a ordem do modelo original, contudo o custo computacional de execução do algoritmo também se eleva. O processamento das simulações variou de alguns segundos (para 10 e 100 simulações) a cerca de 30 min, como podemos registrar em 1000 simulações e $n = 1000$ de séries temporais especificadas para o modelo *AR(2)*.

Observamos ainda que quando os parâmetros dos modelos se aproximam de 0.5 a performance da função aumenta, gerando maiores frequências de séries correspondentes aos modelos inicialmente propostos, independentemente do tamanho da série.

Portanto, podemos concluir que, as simulações são importantes formas de se estudar fatos cujos dados reais estão indisponíveis ou cujos fenômenos só podem ser estudados experimentalmente e que, para o estudo das séries estacionárias, especialmente para os modelos Autorregressivos de ordem 1 (*AR(1)*) Autorregressivos de ordem 2 (*AR(2)*), assim como, os modelos de Médias Móveis de ordem 1 (*MA(1)*) e de Médias Móveis de ordem 2 (*MA(2)*), a função *auto.arima* do Software R, pode ser utilizada na simulação de tais modelos, contudo, deve-se dar preferência a simulações de séries com um número elevado de observações ($n > 30$), e se possível repetir o processo por diversas vezes (em nossos estudos 100 simulações com 100 repetições nos pareceu ser o mais razoável, por apresentar

custo computacional mediano e boa performance na frequência do modelo especificado).

Recomendamos, que a posteriori as séries temporais geradas sejam avaliadas através da função *auto.arima*, assim como, seja analisado os respectivos correlogramas, entre outros testes.

Sugerimos para estudos futuros que esse experimento seja repetido utilizado-se outros Geradores de Números Pseudo Aleatórios (GNPA's) disponíveis no Software R, os quais foram estudados de forma preliminar no decorrer do trabalho em epígrafe.

Referências

ABREU, A. M. M.; RANGEL, J. J. d. A. Simulação computacional: Uma abordagem introdutória. 1999. Disponível em: <http://essentiaeditora.iff.edu.br/index.php/vertices/article/view/1809-2667.19990005/162>. Acesso em: 24 set. 2014. Citado na página 24.

ATHANASOPOULOS, R. J. H. with contributions from G. et al. *forecast: Forecasting functions for time series and linear models*. [S.l.], 2014. R package version 5.5. Disponível em: <http://CRAN.R-project.org/package=forecast>. Citado 3 vezes nas páginas 32, 33 e 50.

BEZERRA, P. M. I. S. *Apostila de Análise de Séries Temporais*. 2006. Disponível em: <http://www.economia.esalq.usp.br/~vitor/series_temporais/apostilas/5515941-Apostila-Series-Temporais_UNESP.pdf>. Acesso em: 24 set. 2014. Citado na página 19.

EHLERS, R. S. Análise de séries temporais. 2003. Acesso em: 14 dez. 2014. Citado 8 vezes nas páginas 12, 13, 15, 16, 17, 20, 21 e 22.

FERREIRA, E. B.; OLIVEIRA, M. S. d. *Introdução à Estatística Básica com R*. 2008. Disponível em: <http://www.quintiliano.prof.ufu.br/index_arquivos/EBR.pdf>. Acesso em: 18 jan. 2014. Citado na página 31.

GOMES, M. I. Simulação e estatística. 1994. Acesso em: 14 dez. 2014. Citado 5 vezes nas páginas 12, 25, 26, 30 e 63.

GUJARATI, D. N. *Econometria Básica*. 4. ed. Rio de Janeiro: Elsevier, 2006. Citado 3 vezes nas páginas 19, 23 e 24.

LANDEIRO, V. L. *Introdução ao uso do programa R*. 2011. Disponível em: <http://cran.r-project.org/doc/contrib/Landeiro-Introducao.pdf>. Acesso em: 29 dez. 2014. Citado na página 31.

METROPOLIS; ULAM, S. *The Monte Carlo Method*. 1949. Acesso em: 27 dez. 2014. Citado na página 28.

MORETTIN, P. A.; TOLOI, C. M. *Modelos para previsão de séries temporais*. 1. ed. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 1981. Citado 3 vezes nas páginas 14, 15 e 16.

MORETTIN, P. A.; TOLOI, C. M. *Análise de Séries Temporais*. 2. ed. São Paulo: E. Blucher, 2006. Citado na página 18.

NASSER, R. B. Mcloud service framwork: arcabouço para desenvolvimentos de serviços baseados na simulação de monte carlo. Rio de Janeiro, 2012. Citado 2 vezes nas páginas 29 e 30.

OLIVEIRA, S. d. Um estudo em séries temporais na análise da receita nominal de vendas de veículos e motos. 2012. Disponível em: <http://periodicos.uniformg.edu.br:21011/periodicos/index.php/testeconexaociencia/article/view/157>. Acesso em: 24 dez. 2014. Citado na página 18.

PETERNELLI, L. F.; MELLO, M. P. *Conhecendo o R: uma visão estatística. Série Didática*. Viçosa-MG: UFV, 2011. Citado 2 vezes nas páginas 31 e 32.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2014. Disponível em: <http://www.R-project.org/>. Citado 2 vezes nas páginas 31 e 65.

REIS, M. M. *Notas de Aula da disciplina Estatística para Administradores*. Curitiba: [s.n.], 2014. Disponível em: <http://www.inf.ufsc.br/~marcelo/Cap4.pdf>. Acesso em: 27 dez. 2014. Citado na página 18.

RIHBANE, F. E. C. Preenchimento de falhas aleatórias de séries temporais micrometeorológicas pela técnica de monte carlo. 2014. Disponível em: <http://www.pgfa.ufmt.br/index.php?option=com_docman&task=doc_download&gid=295&Itemid=236>. Acesso em: 27 dez. 2014. Citado 2 vezes nas páginas 28 e 29.

SOUZA, J. C. U. I. d. Avaliação de títulos conversíveis com opções de compra e venda implícitas em contrato. Rio de Janeiro, 2006. Citado na página 29.

STOCK, J. H.; WATSON, M. W. *Econometria*. 1. ed. São Paulo: Addison Wesley, 2004. Citado na página 18.

VOSS, J. Math5835 - statistical computing. 2011. Disponível em: <http://www1.maths.leeds.ac.uk/~voss/2011/5835/notes.pdf>. Acesso em: 14 dez. 2014. Citado 7 vezes nas páginas 12, 24, 25, 26, 27, 28 e 29.

Anexos

Anexo I

A.1 Geração de NPA's

A.1.1 Uniformes

A.1.1.1 Método Congruencial de Lehmer

Inteiros iniciais: $Z_0, a, b \neq 0, m$

Método de geração: $Z_i \equiv aZ_{i-1} + b(mod\ m), i \geq 1;$

ou $Z_i \equiv aZ_{i-1} + b - \left[\frac{aZ_{i-1}+b}{m}\right] m, i \geq 1;$

ou $Z_i = AMOD(aZ_{i-1} + b, m), i \geq 1;$

NPA's: $R_i = \frac{Z_i}{m}, i \geq 1;$

Escolha das Constantes: A escolha usual de m é em computadores binários $m = 2^\beta$ pois assim facilitamos a *redução módulo m*, que é feita por *truncatura* retendo só os β bits de mais baixa ordem (i.e. mais à direita), e a *divisão* de $Z_i, i \geq 1$, por m envolve unicamente o *deslocamento* do ponto binário β posições para a esquerda e um deslocamento é na realidade muito mais rápido do que uma divisão (m deve ser obviamente ser inferior ao tamanho máximo da palavra do computador que estamos a utilizar).

As restantes constantes são escolhidas com base no *Teorema de Greenberg, Hull e Dobell*: a sucessão $\{Z_n\}_n \geq 1$ tem período máximo m se e somente se (1) b e m forem primos entre si; (2) a_1 for múltiplo de todo primo que divide m; (3) a_1 for múltiplo de 4, se 4 dividir m.

Pode-se pois escolher em máquina binária: $m = 2^\beta$, β elevado; $Z_0 = 0$; b ímpar, qualquer; $a = 1 + 2^{[\beta/2]+1}$.

Algoritmo (num único passo):

X e Y elementos seguintes das sucessões $\{X_n\}_n \geq 0$ e $\{Y_n\}_n \geq 0$ respectivamente

Extraia $j := |kY/m'|, m'$ módulo de $\{Y_n\}_n \geq 0$ (j é um Número Aleatório em $[0, k)$ determinado por Y), e faça $Z := V(j), V(j) := X$.

A.1.2 Geração de NPA's com Distribuição F(.)

O *Algoritmo de Transformação Inversa* (algoritmo universal) é baseado no seguinte argumento probabilístico: Se R for uma NPA uniforme em $(0,1)$ e $F(x)$ uma f.d. arbitrária, então $f^{-1}(u) = inf\{y : F(y) \geq u\}$ o NPA $X = F^{-1}(R)$ tem f.d. $F(x)$.

A.1.3 Exemplos:

No que se segue designaremos por $\{R_i\}_i \geq 1$ (ou genericamente por R) NPA's uniformes em $(0,1)$.

1. Geração de uma NPA *Uniforme* (a,b): $X = a + (b-a)R$

2. Geração de uma NPA *Exponencial* (λ, δ) : $X = \lambda - \delta log\ R$
(métodos de transformação, que não são mais que o algoritmo de transformação inversa).

3. Geração de uma NPA *Normal* (μ, σ) (método de transformação):

Algoritmo de Box-Muller: (a geração tem de ser feita aos pares):

Dados R_1 e R_2 uniformes $(0,1)$ e independentes,

$X_1 = \mu + \sigma\sqrt{-2logR_1}\ cos(2\pi R_2)$, e

$X_2 = \mu + \sigma\sqrt{-2logR_1}\ sin(2\pi R_2)$

São observação independentes de uma Normal (μ, σ).

4. Geração de um NPA $Beta(p,q)$ (método de rejeição):

Dados R_1 e R_2 uniformes em $(0,1)$ e independentes,

faça-se $Y = R_1^{1/p}$, $Z = R_1^{1/q}$.

Se $Y + Z \leq 1$ *então* $X = \dfrac{y}{Y+Z}$ tem f.d. $Beta(p,q)$

5. Geração de um NPA $Gama(\alpha)$:

Faça-se $k = |\alpha|,\ \gamma = \alpha - |\alpha|$

Seja Y um NPA $Beta(\gamma, 1-\gamma)$ $0 < \gamma < 1$ e Z uma exponencial independente de Y.

Então $W = YZ$ é uma $Gama(\gamma)$.

$X = -\sum_{i=1}^{k} logR_i + W$ é uma NPA $Gama(\alpha)$.

6. Geração de um NPA $Binomial(n, p)$:

Como $X = \sum_{i=1}^{n} Y_i,\ Y_i = \begin{cases} 0, & 1-p \\ 1, & p \end{cases}$

Faça-se $X := 0$

Para i desde 1 até n, faça-se $X := X + 1$ se $R_j < p$.

7. Geração de um NPA *poisson(λ)*:

Dados X_i, $i \geq 1$, NPA's exponenciais unitários independentes

$N = maior\ inteiro\ tal\ que\ X = \sum_{i=1}^{N} Y_i < \lambda$ é $Poisson(\lambda)$.

O conteúdo do Anexo acima foi disponibilizado por (GOMES, 1994).

Anexo II

A.2 Tipos de Geradores de Números Aleatórios no R

Os seguintes Geradores de números podem ser utilizados no Software R

Wichmann-Hill: é um vetor de comprimento 3, em que cada $r[i]$ is in $1 : (p[i] - 1)$, em que p é um vetor de igual cumprimento contendo 3 números primos, $p = (30269, 30307, 30323)$. O gerador Wichmann-Hill tem um comprimento de ciclo de $6.9536e^{12}$ $[= prod(p - 1)/4]$.

Marsaglia-Multicarry: é um GNPA recomendado por George Marsaglia na sua lista de discussão 'sci.stat.math'. Tem um período de mais do que 2^{60} e passou em todos os testes (de acordo com o próprio Marsaglia).

Super-Duper: famoso Super-Duper de Marsaglia dos anos 70. Esta é a versão original, que *não* passou no teste MTUPLE da "Diehard battery". Tem um período de cerca de $4,6 * 10^{18}$, para a maioria das sementes iniciais. A semente é do tipo dois inteiros (todos os valores permitidos para a primeira semente: o segundo deve ser ímpar).

Mersenne-Twister: a partir de Matsumoto e Nishimura (1998). É um gerador de números aleatórios com o período de $2^{19937-1}$ e equidistribuição em 623 dimensões consecutivas (em todo o período). A "semente" é um conjunto de 624-dimensões de inteiros de 32 bits, mais uma posição atual naquele conjunto. Esse é gerador utilizado como padrão no R, quando não definido pelo usuário.

Knuth-TAOCP-2002: é um gerador de números aleatórios de 32 bits que usa sequências defasadas de Fibonacci. Ou seja, a recorrência usada é $X[j] = (X[j - 100] - X[j - 37])$ mod 2^{30} e a semente é o conjunto dos 100 últimos números (efectivamente registados como 101 números, sendo o último uma mudança cíclica do buffer). O período é de cerca de 2^{129}.

Knuth-TAOCP: uma versão anterior do Knuth (1997). Teve a inicialização alterada e sua inicialização é feita no código do R interpretado e assim leva um tempo mais curto, porém perceptível.

L'Ecuyer-CMRG: é um 'gerador combinado de múltiplo recursivo' de L'Ecuyer (1999), cada elemento que é um gerador de retorno multiplicativo com três elementos de números inteiros: assim, a semente é um vetor (sinalizado) de inteiros de comprimento 6. O período é de cerca de 2^{191}. Os seis elementos da semente são internamente

considerado como de 32 bits inteiros sem sinal. Nem os três primeiros, nem os três últimos podem ser todos zero, e eles estão limitados a menos de 4294967087 e 4294944443 respectivamente. Isto não é particularmente interessante por si só, mas fornece a base para os vários fluxos usados no pacote paralelo.

Existe ainda, no R, outros Geradores de Números Pseudo-aleatórios fornecidos por usuários, os quais podemos citar: “Kinderman-Ramage”, “Buggy Kinderman-Ramage”, “Ahrens-Dieter”, “Box-Muller” e “Inversion”.

O texto acima é uma reprodução encontrada no Manual do R-Project elaborado por: (R Core Team, 2014).